



Nama: **Adin Adry Tjindarbumi (122140024)**
Environment untuk Multimedia

Tugas Ke: **Worksheet 1: Setup Python**

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 \LaTeX

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 \LaTeX 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: **[UV]**

```

PS D:\tugas1> uv venv multimedia-uv-adin
Using CPython 3.13.7 interpreter at: C:\Users\ASUS\AppData\Local\Programs\Python\Python313\python.exe
Creating virtual environment at: multimedia-uv-adin
Activate with: multimedia-uv-adin\Scripts\activate
PS D:\tugas1> activate
activate : The term 'activate' is not recognized as the name of a cmdlet, function, script file, or operable program.
Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ activate
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (activate:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS D:\tugas1> multimedia-uv-adin\Scripts\activate
(multimedia-uv-adin) PS D:\tugas1>

```

Gambar 1: Perintah aktivasi environment uv

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

```
1 pip install librosa soundfile scipy
2 pip install opencv-python pillow scikit-image matplotlib
3 pip install moviepy
4 pip install numpy pandas jupyter
```

- Screenshot proses instalasi atau output sukses

```
PS D:\tugas1> multimedia-uv-adin\Scripts\activate
(multimedia-uv-adin) PS D:\tugas1> pip install librosa soundfile scipy
Collecting librosa
  Downloading librosa-0.11.0-py3-none-any.whl.metadata (8.7 kB)
Collecting soundfile
  Downloading soundfile-0.13.1-py2.py3-none-win_amd64.whl.metadata (16 kB)
Collecting scipy
  Downloading scipy-1.16.1-cp313-cp313-win_amd64.whl.metadata (60 kB)
```

Gambar 2: Perintah instalasi library audio

```
(multimedia-uv-adin) PS D:\tugas1> pip install opencv-python pillow scikit-image matplotlib
Collecting opencv-python
  Downloading opencv_python-4.12.0.88-cp37-abi3-win_amd64.whl.metadata (19 kB)
Collecting pillow
  Downloading pillow-11.3.0-cp313-cp313-win_amd64.whl.metadata (9.2 kB)
Collecting scikit-image
  Downloading scikit_image-0.25.2-cp313-cp313-win_amd64.whl.metadata (14 kB)
Collecting matplotlib
  Downloading matplotlib-3.10.5-cp313-cp313-win_amd64.whl.metadata (11 kB)
Requirement already satisfied: numpy<2.3.0,>=2 in c:\users\asus\appdata\local\programs\python\python313\lib\site-packages (from opencv-python) (2.2.6)
Requirement already satisfied: scipy>=1.11.4 in c:\users\asus\appdata\local\programs\python\python313\lib\site-packages (from scikit-image) (1.16.1)
Collecting networkx>=3.0 (from scikit-image)
  Downloading networkx-3.5-py3-none-any.whl.metadata (6.3 kB)
Collecting imageio!=2.35.0,>=2.33 (from scikit-image)
  Downloading imageio-2.37.0-py3-none-any.whl.metadata (5.2 kB)
Collecting tifffile>=2022.8.12 (from scikit-image)
```

Gambar 3: Perintah instalasi library image

```
(multimedia-uv-adin) PS D:\tugas1> pip install moviepy
Collecting moviepy
  Downloading moviepy-2.2.1-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: decorator<6.0,>=4.0.2 in c:\users\asus\appdata\local\programs\python\python313\lib\site-packages (from moviepy) (5.2.1)
Requirement already satisfied: imageio<3.0,>=2.5 in c:\users\asus\appdata\local\programs\python\python313\lib\site-packages (from moviepy) (2.37.0)
Collecting imageio_ffmpeg>=0.2.0 (from moviepy)
  Downloading imageio_ffmpeg-0.6.0-py3-none-win_amd64.whl.metadata (1.5 kB)
Requirement already satisfied: numpy>=1.25.0 in c:\users\asus\appdata\local\programs\python\python313\lib\site-packages (from moviepy) (2.2.6)
Collecting proglog<=1.0.0 (from moviepy)
  Downloading proglog-0.1.12-py3-none-any.whl.metadata (794 bytes)
Collecting python-dotenv>=0.10 (from moviepy)
  Downloading python_dotenv-1.1.1-py3-none-any.whl.metadata (24 kB)
Requirement already satisfied: pillow<12.0,>=9.2.0 in c:\users\asus\appdata\local\programs\python\python313\lib\site-pac
```

Gambar 4: Perintah instalasi library video

- Daftar library yang berhasil diinstall dengan versinya

```
(multimedia-uv-adin) PS D:\tugas1> pip list
Package            Version
-----
audioop-lts        0.2.2
audioread           3.0.1
certifi             2025.8.3
cffi                1.17.1
charset-normalizer  3.4.3
colorama            0.4.6
contourpy           1.3.3
cyclor              0.12.1
decorator           5.2.1
fonttools           4.59.2
idna                3.10
imageio             2.37.0
imageio-ffmpeg      0.6.0
joblib              1.5.2
kiwisolver          1.4.9
lazy_loader         0.4
librosa             0.11.0
llvmlite            0.44.0
matplotlib          3.10.5
moviepy             2.2.1
msgpack             1.1.1
networkx            3.5
numba               0.61.2
numpy               2.2.6
opencv-python       4.12.0.88
packaging           25.0
pillow              11.3.0
pip                25.2
platformdirs        4.4.0
pooch               1.8.2
proglog             0.1.12
pycparser           2.22
pyparsing           3.2.3
python-dateutil     2.9.0.post0
python-dotenv       1.1.1
requests            2.32.5
scikit-image        0.25.2
scikit-learn        1.7.1

scipy               1.16.1
six                 1.17.0
soundfile           0.13.1
soxr                0.5.0.post1
standard-aifc       3.13.0
standard-chunk      3.13.0
standard-sunau      3.13.0
threadpoolctl       3.6.0
tiffiff             2025.8.28
tqdm                4.67.1
typing_extensions   4.15.0
urllib3             2.5.0
uv                  0.8.13
(multimedia-uv-adin) PS D:\tugas1> |
```

Gambar 5: Daftar library yang berhasil diinstall

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

```

PS D:\tugas1\multimedia-uv-adin> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe d:/tugas1/multimedia-uv-ad
in/test_audio.py
Generated 2s sine wave at 440Hz
Sample rate: 44100Hz
Total samples: 88200
PS D:\tugas1\multimedia-uv-adin>

```

Gambar 6: Output test audio processing

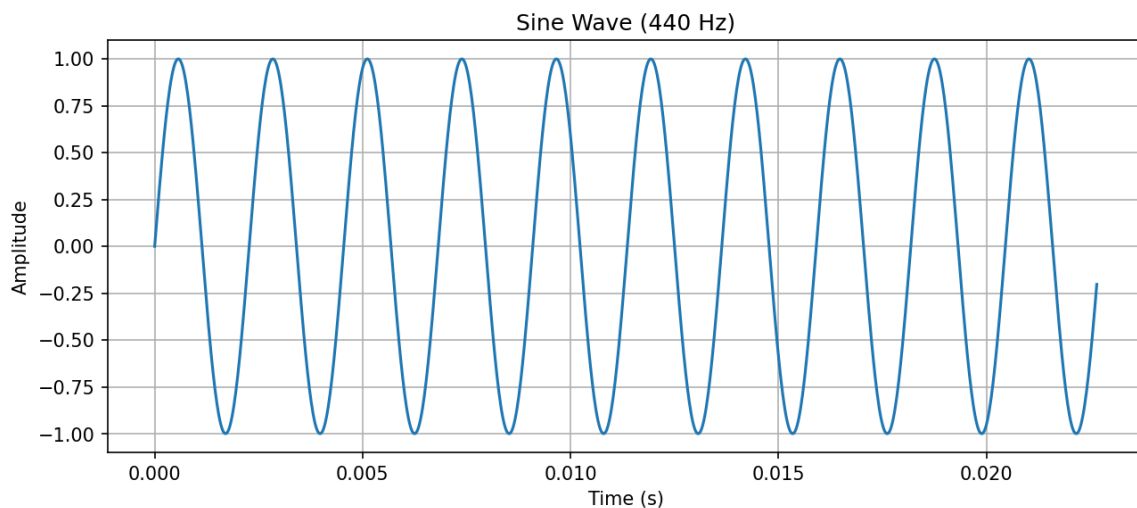
```

PS D:\tugas1\multimedia-uv-adin> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe d:/tugas1/multimedia-uv-ad
in/test_image.py
Created test image: 400x300 pixels
Image shape: (300, 400, 3)
Image dtype: uint8

```

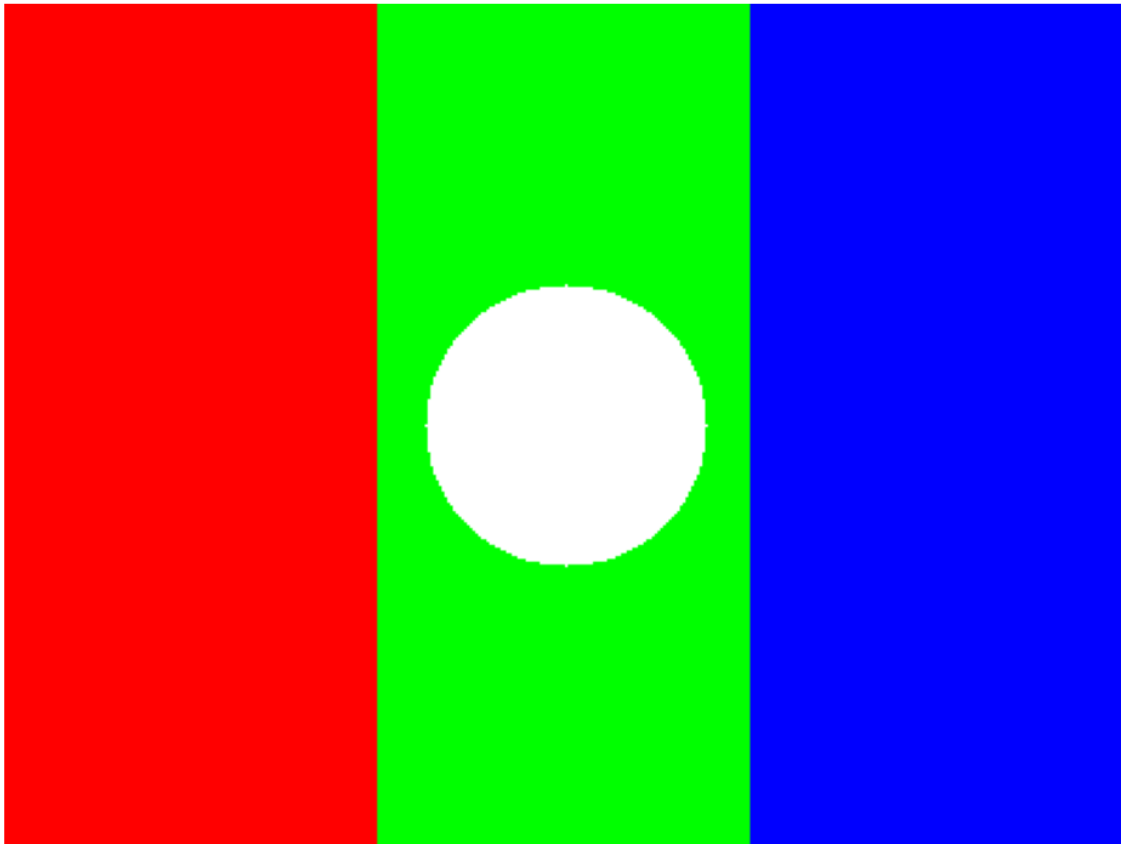
Gambar 7: Output test image processing

- Gambar yang dihasilkan (sine_wave_test.png dan test_image.png)



Gambar 8: Output gambar sine wave dari test audio processing

Test Image with RGB Stripes and White Circle



Gambar 9: Output gambar test image processing

- Tidak ada error message

4 Bagian Laporan

4.1 Output Verifikasi Instalasi

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

```
1 #OUTPUT test_audio.py
2 PS D:\tugas1\multimedia-uv-adin> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe
   d:/tugas1/multimedia-uv-adin/test_audio.py
3 Generated 2s sine wave at 440Hz
4 Sample rate: 44100Hz
5 Total samples: 88200
6
7 # OUTPUT test_image.py
8 PS D:\tugas1\multimedia-uv-adin> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe
   d:/tugas1/multimedia-uv-adin/test_image.py
9 Created test image: 400x300 pixels
10 Image shape: (300, 400, 3)
11 Image dtype: uint8
```

Kode 10: Output verifikasi instalasi

4.2 Screenshot Hasil Test

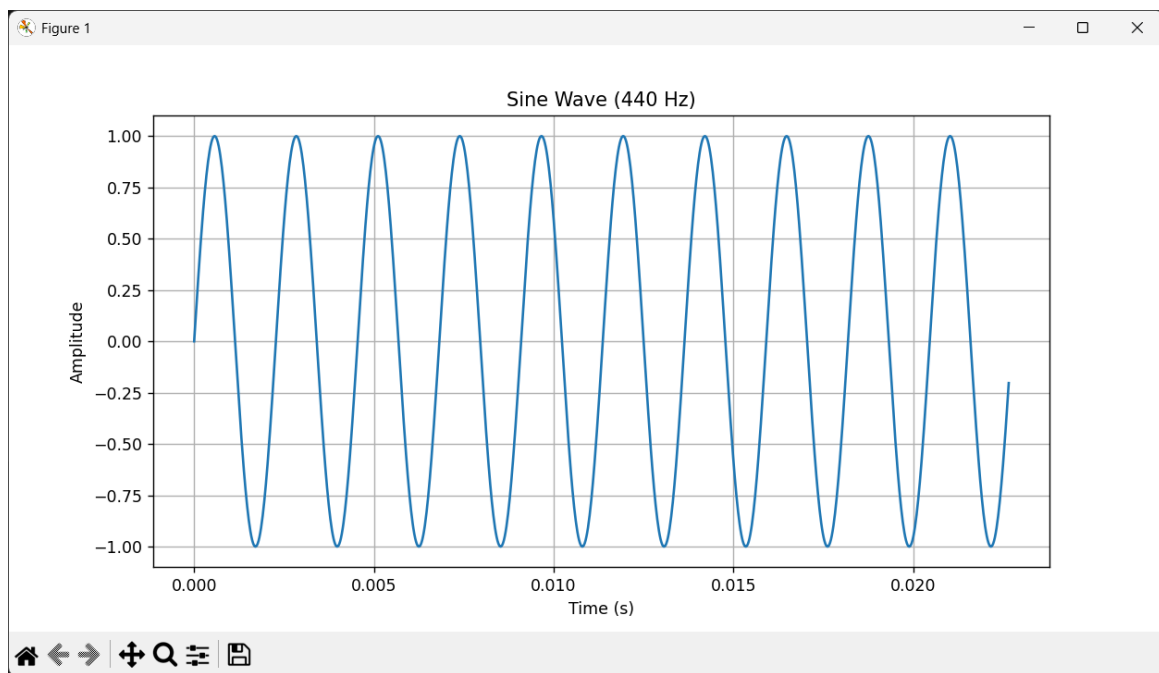
Sisipkan screenshot atau gambar hasil dari:

- Terminal/command prompt yang menunjukkan environment aktif

```
(multimedia-uv-adin) PS D:\tugas1> multimedia-uv-adin\Scripts\activate  
(multimedia-uv-adin) PS D:\tugas1> |
```

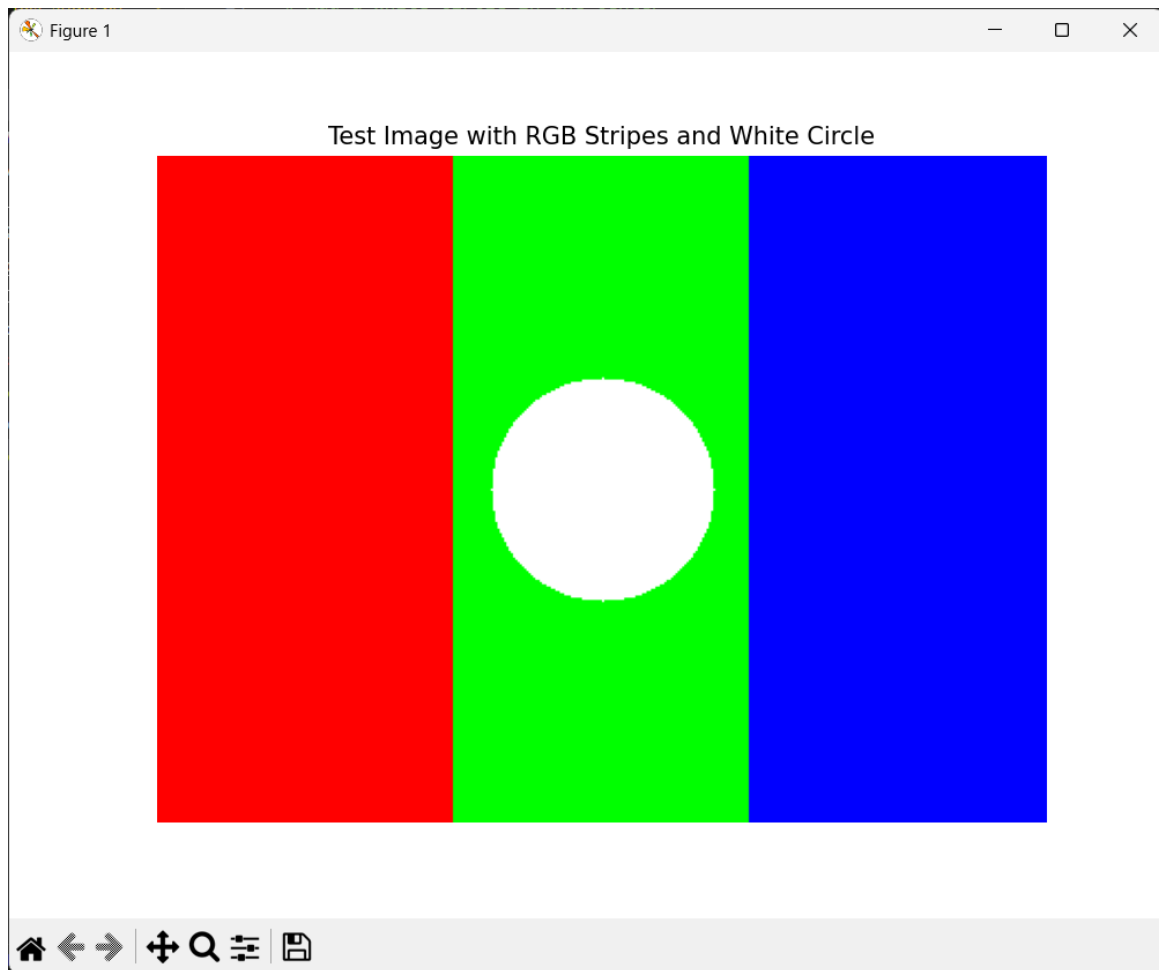
Gambar 10: Environment uv aktif

- Output dari script test audio (sine wave plot)



Gambar 11: Output test audio processing

- Output dari script test image (RGB stripes dengan circle)



Gambar 12: Output test image processing

4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

1. Mengapa penting menggunakan environment terpisah untuk project multimedia?

Karena tiap project multimedia biasanya membutuhkan library yang berbeda-beda, sehingga dengan menggunakan environment yang terpisah, kita dapat meminimalisir potensi terjadinya konflik library yang bertabrakan.

2. Apa perbedaan utama antara conda, venv, dan uv? Mengapa Anda memilih tool yang Anda gunakan?

Perbedaan yang mencolok antara conda, venv, dan uv menurut saya adalah conda lebih lengkap karena memiliki package manager tetapi ukurannya lebih besar, sedangkan venv lebih ringan dan merupakan tool bawaan python tetapi tidak memiliki package manager. Sedangkan uv lebih modern dan cepat dibanding dengan venv, serta memiliki fitur-fitur tambahan seperti manajemen dependency yang lebih baik. Alasan utama saya memilih uv adalah karena ukurannya yang ringan dan sangat cepat dibandingkan conda dan venv.

3. Library mana yang paling sulit diinstall dan mengapa?

Menurut saya, tidak ada library yang sulit untuk diinstall karena saat penginstalan tidak ada error yang muncul dan semua berjalan dengan lancar.

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

Berdasarkan yang saya ketahui, caranya adalah dengan membuat environment baru dan apabila perlu menginstall kembali library yang diperlukan satu persatu.

5. Jelaskan fungsi dari masing-masing library yang berhasil Anda install!

Berikut fungsi dari masing-masing library yang berhasil saya install:

- **Librosa:** Library untuk analisis dan pemrosesan audio, seperti ekstraksi fitur audio.
- **Soundfile:** Library untuk membaca dan menulis file audio.
- **Scipy:** Library untuk komputasi ilmiah, termasuk pemrosesan sinyal.
- **OpenCV:** Library untuk pemrosesan gambar dan video.
- **Pillow:** Library untuk manipulasi gambar.
- **Scikit-image:** Library untuk pemrosesan gambar berbasis NumPy.
- **Matplotlib:** Library untuk visualisasi data, termasuk plotting grafik.
- **MoviePy:** Library untuk pengeditan video.
- **Numpy:** Library untuk komputasi numerik dengan array multidimensi.
- **Pandas:** Library untuk manipulasi dan analisis data.
- **Jupyter:** Alat untuk membuat dan berbagi dokumen yang berisi kode, visualisasi, dan teks.

4.4 Troubleshooting

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

- **Masalah 1:** Ketika saya ingin menjalankan script test audio dan image, muncul error `ModuleNotFoundError: No module named 'numpy'`.

Solusi: Setelah saya cek, modul numpy sudah terinstall di environment uv saya. Ternyata masalahnya adalah saya lupa mengaktifkan environment uv sebelum menjalankan script. Setelah saya aktifkan environment uv, script dapat berjalan dengan lancar tanpa error.

5 Export Environment untuk Reproduksi

Sebagai langkah terakhir, export environment Anda agar dapat direproduksi:

```
1 audioop-lts==0.2.2
2 audioread==3.0.1
3 certifi==2025.8.3
4 cffi==1.17.1
5 charset-normalizer==3.4.3
6 colorama==0.4.6
7 contourpy==1.3.3
8 cycler==0.12.1
9 decorator==5.2.1
10 fonttools==4.59.2
11 idna==3.10
12 imageio==2.37.0
13 imageio-ffmpeg==0.6.0
14 joblib==1.5.2
15 kiwisolver==1.4.9
16 lazy_loader==0.4
```

```
17 librosa==0.11.0
18 llvmlite==0.44.0
19 matplotlib==3.10.5
20 moviepy==2.2.1
21 msgpack==1.1.1
22 networkx==3.5
23 numba==0.61.2
24 numpy==2.2.6
25 opencv-python==4.12.0.88
26 packaging==25.0
27 pillow==11.3.0
28 platformdirs==4.4.0
29 pooch==1.8.2
30 proglog==0.1.12
31 pycparser==2.22
32 pyparsing==3.2.3
33 python-dateutil==2.9.0.post0
34 python-dotenv==1.1.1
35 requests==2.32.5
36 scikit-image==0.25.2
37 scikit-learn==1.7.1
38 scipy==1.16.1
39 six==1.17.0
40 soundfile==0.13.1
41 soxr==0.5.0.post1
42 standard-aifc==3.13.0
43 standard-chunk==3.13.0
44 standard-sunau==3.13.0
45 threadpoolctl==3.6.0
46 tiffio==2025.8.28
47 tqdm==4.67.1
48 typing_extensions==4.15.0
49 urllib3==2.5.0
50 uv==0.8.13
```

Kode 11: Environment/Requirements file

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

Pengalaman saya dalam melakukan setup Python environment untuk multimedia ini sangat menyenangkan dan memberikan banyak wawasan baru. Saya belajar tentang pentingnya manajemen environment untuk menghindari konflik antar library. Proses instalasi library-library multimedia juga berjalan lancar tanpa kendala yang besar, untuk persiapan project multimedia selanjutnya, saya merasa lebih siap karena sudah memahami cara mengatur environment dan menginstall library yang diperlukan. Saran saya untuk mahasiswa lain yang akan melakukan setup serupa adalah untuk selalu memastikan environment sudah aktif sebelum menjalankan script, serta mendokumentasikan setiap langkah yang dilakukan agar mudah untuk direproduksi di project selanjutnya.

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

References

- [1] OpenAI, “Percakapan chatgpt mengenai perbedaan conda, venv, dan uv,” <https://chatgpt.com/share/68b1602e-a83c-8012-b54e-21c1c71b5368>, 18 Dec. 2025, accessed: 2025-8-29.