Program Studi Teknik Informatika
Institut Teknologi Sumatera

Nama: **Nashwa Putri Laisya (122140180)** Tugas Ke: **Worksheet 1: Setup Python Environment untuk Multimedia**
Mata Kuliah: **Sistem Teknologi Multimedia (IF25-40305)** Tanggal: August 28, 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:

```
# Membuat environment baru dengan nama 'multimedia'
conda create -n multimedia python=3.11

# Mengaktifkan environment
conda activate multimedia

# Verifikasi environment aktif
conda info --envs
```

Kode 1: Membuat environment dengan Conda

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

```
# Membuat environment baru
python3 -m venv multimedia-env

# Mengaktifkan environment (Linux/Mac)
source multimedia-env/bin/activate

# Mengaktifkan environment (Windows)
# multimedia-env\Scripts\activate

# Verifikasi environment aktif
which python
```

Kode 2: Membuat environment dengan venv

3.2.3 Opsi 3: Menggunakan uv (Modern dan cepat)

```
# Install uv terlebih dahulu jika belum ada
# pip install uv

# Membuat environment baru
uv venv multimedia-uv

# Mengaktifkan environment (Linux/Mac)
source multimedia-uv/bin/activate

# Mengaktifkan environment (Windows)
# multimedia-uv\Scripts\activate

# Verifikasi environment aktif
which python
```

Kode 3: Membuat environment dengan uv

Dokumentasi:

- Tool manajemen environment yang saya pilih: **uv**
- Screenshot atau copy-paste output dari perintah verifikasi environment

```
nashwalaisya@192 ~ % python3 -m pip install uv
Requirement already satisfied: uv in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (0.8.13)
nashwalaisya@192 ~ % uv venv multimedia-uv
Using CPython 3.13.7 interpreter at: /Library/Frameworks/Python.framework/Versions/3.13/bin/python3
Creating virtual environment at: multimedia-uv
Activate with: source multimedia-uv/bin/activate
nashwalaisya@192 ~ % source multimedia-uv/bin/activate
(multimedia-uv) nashwalaisya@192 ~ % which python3
/Users/nashwalaisya/multimedia-uv/bin/python3
(multimedia-uv) nashwalaisya@192 ~ % which python
/Users/nashwalaisya/multimedia-uv/bin/python
(multimedia-uv) nashwalaisya@192 ~ %
```

Gambar 1: Screenshot pembuatan environment uv

3.3 Bagian 2: Instalasi Library Multimedia

Setelah environment aktif, install library-library berikut:

3.3.1 Library Audio Processing

```
# Untuk conda:
conda install -c conda-forge librosa soundfile scipy

# Untuk pip (venv/uv):
pip install librosa soundfile scipy
```

Kode 4: Instalasi library audio

3.3.2 Library Image Processing

```
# Untuk conda:
conda install -c conda-forge opencv pillow scikit-image matplotlib

# Untuk pip (venv/uv):
pip install opencv-python pillow scikit-image matplotlib
```

Kode 5: Instalasi library image

3.3.3 Library Video Processing

```
# Untuk conda:
conda install -c conda-forge ffmpeg
pip install moviepy

# Untuk pip (venv/uv):
pip install moviepy
```

Kode 6: Instalasi library video

3.3.4 Library General Purpose

```
# Untuk conda:
conda install numpy pandas jupyter

# Untuk pip (venv/uv):
pip install numpy pandas jupyter
```

Kode 7: Instalasi library umum

Dokumentasi Instalasi Library Audio Processing:

- Perintah instalasi yang saya gunakan

```
python -m pip install librosa soundfile scipy
```

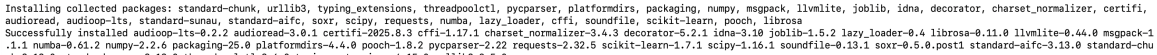
Kode 8: Perintah instalasi library audio processing yang digunakan

- Screenshot proses instalasi atau output sukses

```
(multimedia-uv) nashwalaisya@192 ~ % python -m 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-macosx_11_0_arm64.whl.metadata (16 kB)
Collecting scipy
  Downloading scipy-1.16.1-cp313-cp313-macosx_14_0_arm64.whl.metadata (61 kB)
Collecting audioread>=2.1.9 (from librosa)
  Downloading audioread-3.0.1-py3-none-any.whl.metadata (8.4 kB)
Collecting numba>=0.51.0 (from librosa)
  Downloading numba-0.61.2-cp313-cp313-macosx_11_0_arm64.whl.metadata (2.7 kB)
Collecting numpy>=1.22.3 (from librosa)
  Downloading numpy-2.3.2-cp313-cp313-macosx_14_0_arm64.whl.metadata (62 kB)
Collecting scikit-learn>=1.1.0 (from librosa)
  Downloading scikit_learn-1.7.1-cp313-cp313-macosx_12_0_arm64.whl.metadata (11 kB)
Collecting joblib>=1.0 (from librosa)
  Downloading joblib-1.5.2-py3-none-any.whl.metadata (5.6 kB)
Collecting decorator>=4.3.0 (from librosa)
  Downloading decorator-5.2.1-py3-none-any.whl.metadata (3.9 kB)
Collecting pooch>=1.1 (from librosa)
  Downloading pooch-1.8.2-py3-none-any.whl.metadata (10 kB)
Collecting soxr>=0.3.2 (from librosa)
  Downloading soxr-0.5.0.post1-cp312-abi3-macosx_11_0_arm64.whl.metadata (5.6 kB)
Collecting typing_extensions>=4.1.1 (from librosa)
  Downloading typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
Collecting lazy_loader>=0.1 (from librosa)
  Downloading lazy_loader-0.4-py3-none-any.whl.metadata (7.6 kB)
Collecting msgpack>=1.0 (from librosa)
  Downloading msgpack-1.1.1-cp313-cp313-macosx_11_0_arm64.whl.metadata (8.4 kB)
Collecting standard-aifc (from librosa)
  Downloading standard_aifc-3.13.0-py3-none-any.whl.metadata (969 bytes)
Collecting standard-sunau (from librosa)
  Downloading standard_sunau-3.13.0-py3-none-any.whl.metadata (914 bytes)
Collecting cffi>=1.0 (from soundfile)
  Downloading cffi-1.17.1-cp313-cp313-macosx_11_0_arm64.whl.metadata (1.5 kB)
Collecting pycparser (from cffi>=1.0->soundfile)
  Downloading pycparser-2.22-py3-none-any.whl.metadata (943 bytes)
Collecting packaging (from lazy_loader>=0.1->librosa)
  Downloading packaging-25.0-py3-none-any.whl.metadata (3.3 kB)
Collecting llvmlite<0.45,>=0.44.0dev0 (from numba>=0.51.0->librosa)
  Downloading llvmlite-0.44.0-cp313-cp313-macosx_11_0_arm64.whl.metadata (4.8 kB)
Collecting numpy>=1.22.3 (from librosa)
  Downloading numpy-2.2.6-cp313-cp313-macosx_14_0_arm64.whl.metadata (62 kB)
Collecting platformdirs>=2.5.0 (from pooch>=1.1->librosa)
  Downloading platformdirs-4.4.0-py3-none-any.whl.metadata (12 kB)
Collecting requests>=2.19.0 (from pooch>=1.1->librosa)
  Downloading requests-2.32.5-py3-none-any.whl.metadata (4.9 kB)
Collecting charset_normalizer<4,>=2 (from requests>=2.19.0->pooch>=1.1->librosa)
  Downloading charset_normalizer-3.4.3-cp313-cp313-macosx_10_13_universal2.whl.metadata (36 kB)
Collecting idna<4,>=2.5 (from requests>=2.19.0->pooch>=1.1->librosa)
  Downloading idna-3.10-py3-none-any.whl.metadata (10 kB)
Collecting urllib3<3,>=1.21.1 (from requests>=2.19.0->pooch>=1.1->librosa)
  Downloading urllib3-2.5.0-py3-none-any.whl.metadata (6.5 kB)
Collecting certifi>=2017.4.17 (from requests>=2.19.0->pooch>=1.1->librosa)
  Using cached certifi-2025.8.3-py3-none-any.whl.metadata (2.4 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn>=1.1.0->librosa)
  Downloading threadpoolctl-3.6.0-py3-none-any.whl.metadata (13 kB)
Collecting standard-chunk (from standard-aifc->librosa)
  Downloading standard_chunk-3.13.0-py3-none-any.whl.metadata (860 bytes)
Collecting audioop-lts (from standard-aifc->librosa)
  Downloading audioop_lts-0.2.2-cp313-abi3-macosx_11_0_arm64.whl.metadata (2.0 kB)
Downloading librosa-0.11.0-py3-none-any.whl (260 kB)
Downloading soundfile-0.13.1-py2.py3-none-macosx_11_0_arm64.whl (1.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 2.0 MB/s  0:00:00
Downloading scipy-1.16.1-cp313-cp313-macosx_14_0_arm64.whl (20.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20.8/20.8 MB 3.5 MB/s  0:00:06
Downloading audioread-3.0.1-py3-none-any.whl (23 kB)
Downloading cffi-1.17.1-cp313-cp313-macosx_11_0_arm64.whl (178 kB)
Downloading decorator-5.2.1-py3-none-any.whl (9.2 kB)
Downloading joblib-1.5.2-py3-none-any.whl (308 kB)
Downloading lazy_loader-0.4-py3-none-any.whl (12 kB)
Downloading msgpack-1.1.1-cp313-cp313-macosx_11_0_arm64.whl (78 kB)
Downloading numba-0.61.2-cp313-cp313-macosx_11_0_arm64.whl (2.8 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.8/2.8 MB 3.6 MB/s  0:00:00
Downloading llvmlite-0.44.0-cp313-cp313-macosx_11_0_arm64.whl (26.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 26.2/26.2 MB 3.7 MB/s  0:00:06
Downloading numpy-2.2.6-cp313-cp313-macosx_14_0_arm64.whl (5.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.1/5.1 MB 3.6 MB/s  0:00:01
Downloading pooch-1.8.2-py3-none-any.whl (64 kB)
Downloading packaging-25.0-py3-none-any.whl (66 kB)
Downloading platformdirs-4.4.0-py3-none-any.whl (18 kB)
Downloading requests-2.32.5-py3-none-any.whl (64 kB)
Downloading charset_normalizer-3.4.3-cp313-cp313-macosx_10_13_universal2.whl (205 kB)
Downloading idna-3.10-py3-none-any.whl (70 kB)
Downloading urllib3-2.5.0-py3-none-any.whl (129 kB)
Using cached certifi-2025.8.3-py3-none-any.whl (161 kB)
Downloading scikit_learn-1.7.1-cp313-cp313-macosx_12_0_arm64.whl (8.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.6/8.6 MB 3.6 MB/s  0:00:02
Downloading soxr-0.5.0.post1-cp312-abi3-macosx_11_0_arm64.whl (156 kB)
Downloading threadpoolctl-3.6.0-py3-none-any.whl (18 kB)
Downloading typing_extensions-4.15.0-py3-none-any.whl (44 kB)
Downloading pycparser-2.22-py3-none-any.whl (117 kB)
Downloading standard_aifc-3.13.0-py3-none-any.whl (10 kB)
Downloading audioop_lts-0.2.2-cp313-abi3-macosx_11_0_arm64.whl (26 kB)
Downloading standard_chunk-3.13.0-py3-none-any.whl (4.9 kB)
Downloading standard_sunau-3.13.0-py3-none-any.whl (7.4 kB)
Installing collected packages: standard-chunk, urllib3, typing_extensions, threadpoolctl, pycpar
Successfully installed audioop-lts-0.2.2 audioread-3.0.1 certifi-2025.8.3 cffi-1.17.1 charset_no
 soxr-0.5.0.post1 standard-aifc-3.13.0 standard-chunk-3.13.0 standard-sunau-3.13.0 threadpoolctl
(multimedia-uv) nashwalaisya@192 ~ %
```

Gambar 2: Screenshot instalasi library audio processing

- Daftar library yang berhasil diinstall dengan versinya



Gambar 3: Library audio processing yang berhasil diinstal

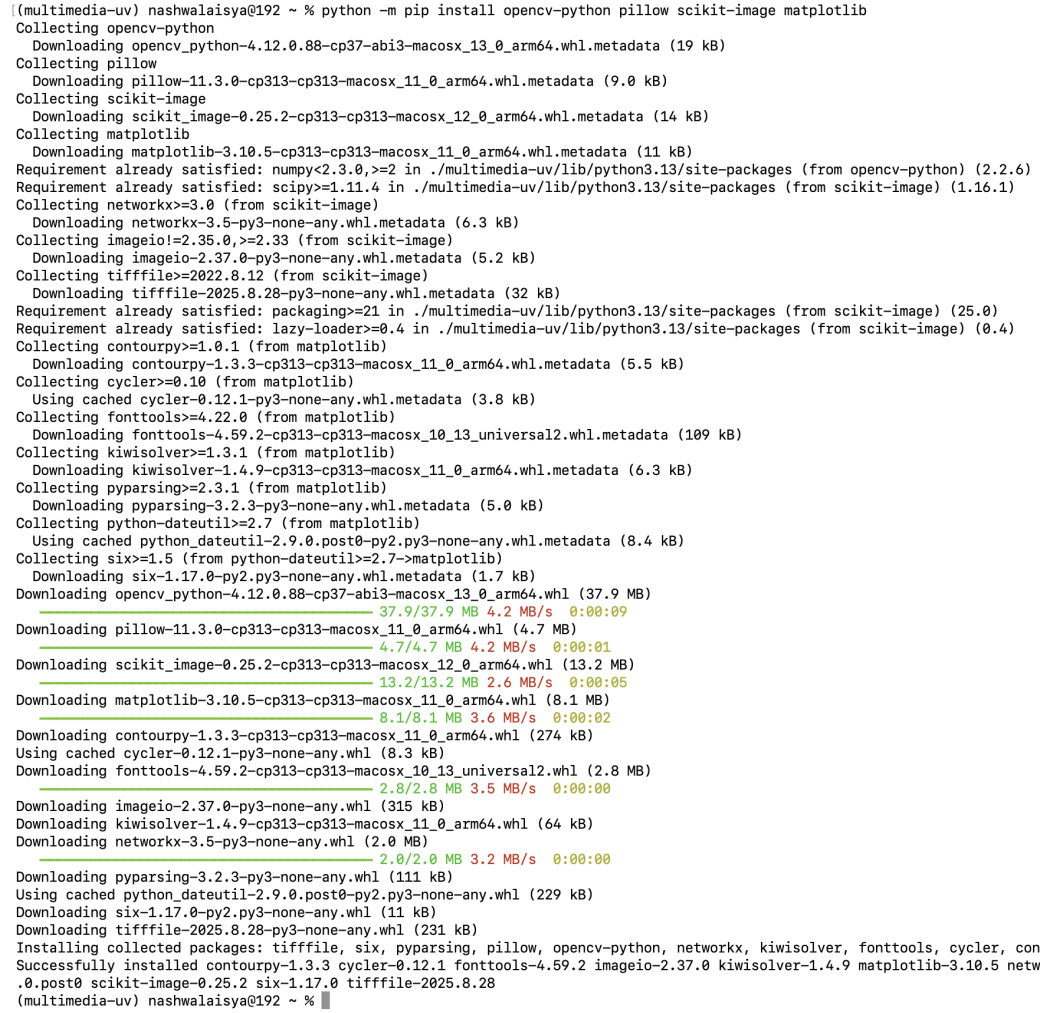
Dokumentasi Instalasi Library Image Processing:

- Perintah instalasi yang saya gunakan

```
python -m pip install opencv-python pillow scikit-image matplotlib
```

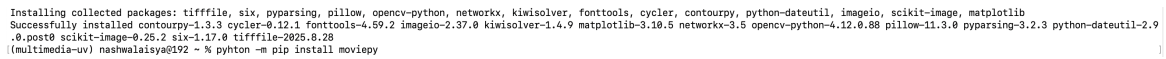
Kode 9: Perintah instalasi library image processing yang digunakan

- Screenshot proses instalasi atau output sukses



Gambar 4: Screenshot instalasi library image processing

- Daftar library yang berhasil diinstall dengan versinya



Gambar 5: Library image processing yang berhasil diinstal

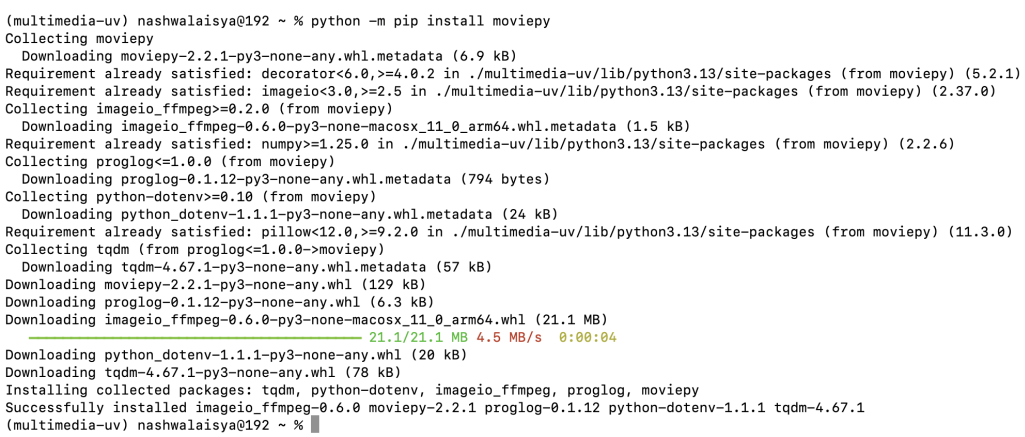
Dokumentasi Instalasi Library Video Processing:

- Perintah instalasi yang saya gunakan

```
1        python -m pip install moviepy
```

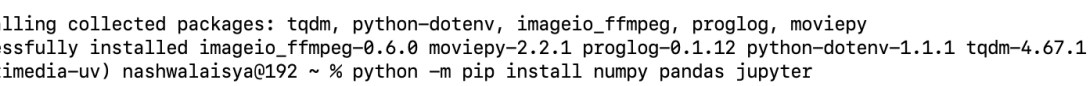
Kode 10: Perintah instalasi library video processing yang digunakan

- Screenshot proses instalasi atau output sukses



Gambar 6: Screenshot instalasi library video processing

- Daftar library yang berhasil diinstall dengan versinya



Gambar 7: Library video processing yang berhasil diinstal

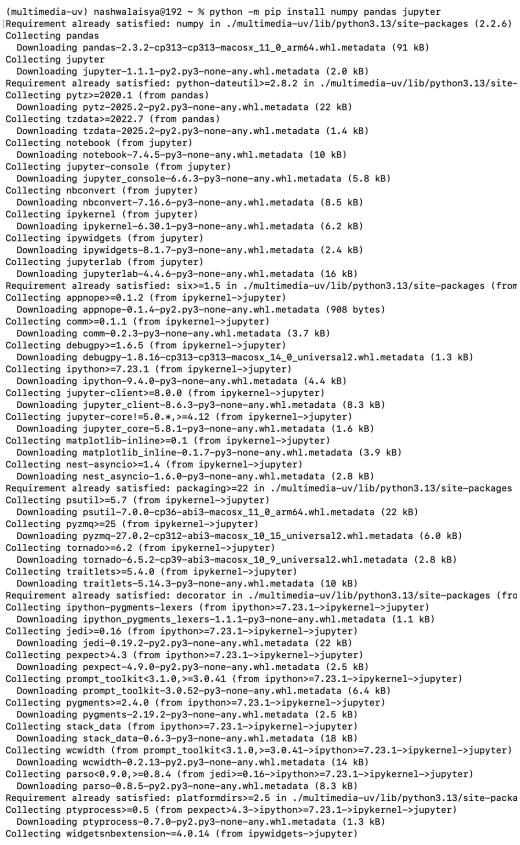
Dokumentasi Instalasi Library General Purpose:

- Perintah instalasi yang saya gunakan

```
1        python -m pip install numpy pandas jupyter
```

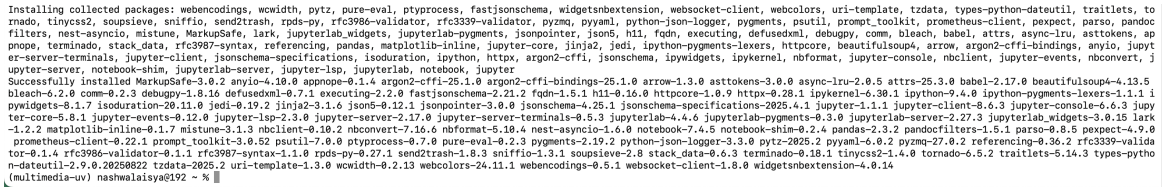
Kode 11: Perintah instalasi library general purpose yang digunakan

- Screenshot proses instalasi atau output sukses



Gambar 8: Screenshot instalasi library general purpose

- Daftar library yang berhasil diinstall dengan versinya



Gambar 9: Library general processing yang berhasil diinstal

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

```
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, int(sample_rate * duration))
audio_signal = np.sin(2 * np.pi * frequency * t)

# Plot waveform
plt.figure(figsize=(10, 4))
plt.plot(t[:1000], audio_signal[:1000])  # Plot first 1000 samples
plt.title('Sine Wave (440 Hz)')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.grid(True)
plt.savefig('sine_wave_test.png', dpi=150, bbox_inches='tight')
plt.show()

print(f"Generated {duration}s sine wave at {frequency}Hz")
print(f"Sample rate: {sample_rate}Hz")
print(f"Total samples: {len(audio_signal)}")
```

Kode 12: Test audio processing sederhana

3.5.2 Test Image Processing

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image

# Create a simple test image
width, height = 400, 300
image = np.zeros((height, width, 3), dtype=np.uint8)

# Add some patterns
image[:, :width//3, 0] = 255  # Red section
image[:, width//3:2*width//3, 1] = 255  # Green section
image[:, 2*width//3:, 2] = 255  # Blue section

# Add a white circle in the center
center_x, center_y = width//2, height//2
radius = 50
Y, X = np.ogrid[:height, :width]
mask = (X - center_x)**2 + (Y - center_y)**2 <= radius**2
image[mask] = [255, 255, 255]

# Display and save
plt.figure(figsize=(8, 6))
plt.imshow(image)
plt.title('Test Image with RGB Stripes and White Circle')
plt.axis('off')
plt.savefig('test_image.png', dpi=150, bbox_inches='tight')
plt.show()

print(f"Created test image: {width}x{height} pixels")
print(f"Image shape: {image.shape}")
print(f"Image dtype: {image.dtype}")
```

Kode 13: Test image processing sederhana

Dokumentasikan hasil eksekusi:

- Screenshot output dari kedua script di atas

- Gambar yang dihasilkan
- Error message jika ada dan cara mengatasinya

4 Bagian Laporan

4.1 Output Verifikasi Instalasi

Perintah yang digunakan:

```
try:
    import librosa
    print("librosa: Success")
except Exception as e:
    print("librosa: ERROR ->", e)

try:
    import soundfile
    print("soundfile: Success")
except Exception as e:
    print("soundfile: ERROR ->", e)

try:
    import scipy
    print("scipy: Success")
except Exception as e:
    print("scipy: ERROR ->", e)

try:
    import cv2
    print("opencv (cv2): Success")
except Exception as e:
    print("opencv: ERROR ->", e)

try:
    from PIL import Image
    print("Pillow (PIL): Success")
except Exception as e:
    print("Pillow: ERROR ->", e)

try:
    import skimage
    print("scikit-image: Success")
except Exception as e:
    print("scikit-image: ERROR ->", e)

try:
    import matplotlib
    print("matplotlib: Success")
except Exception as e:
    print("matplotlib: ERROR ->", e)

try:
    import moviepy
    print("moviepy: Success")
except Exception as e:
    print("moviepy: ERROR ->", e)

try:
    import numpy
    print("numpy: Success")
```

```
except Exception as e:
    print("numpy: ERROR ->", e)

try:
    import pandas
    print("pandas: Success")
except Exception as e:
    print("pandas: ERROR ->", e)
```

Kode 14: Perintah untuk verifikasi instalasi

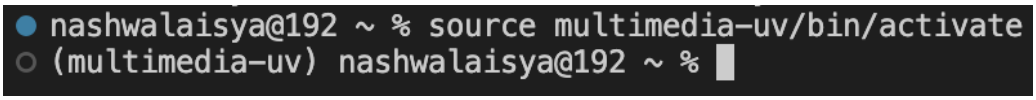
Output

```
(multimedia-uv) nashwalaisya@192 ~ % python "/Users/nashwalaisya/Documents/Semester - 7/Sistem
    Teknologi Multimedia/Worksheet 1/verifikasi_instalasi.py"
librosa: Success
soundfile: Success
scipy: Success
opencv (cv2): Success
Pillow (PIL): Success
scikit-image: Success
matplotlib: Success
moviepy: Success
numpy: Success
pandas: Success
```

Kode 15: Output verifikasi instalasi

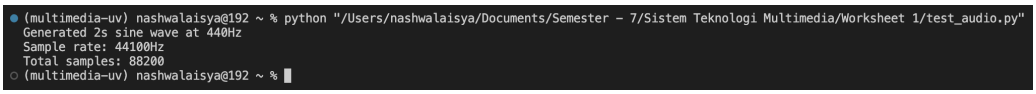
4.2 Screenshot Hasil Test

- Terminal/command prompt yang menunjukkan environment aktif

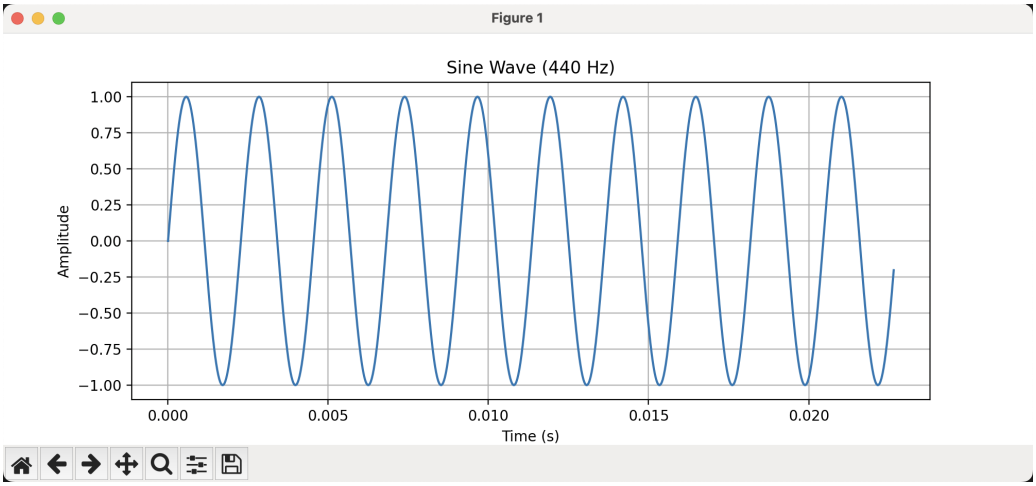


Gambar 10: Aktivasi environment

- Output dari script test audio (sine wave plot)

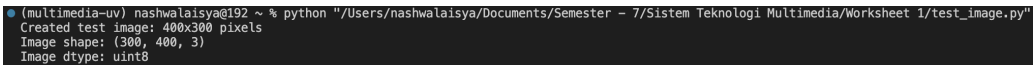


Gambar 11: Output script test audio pada terminal

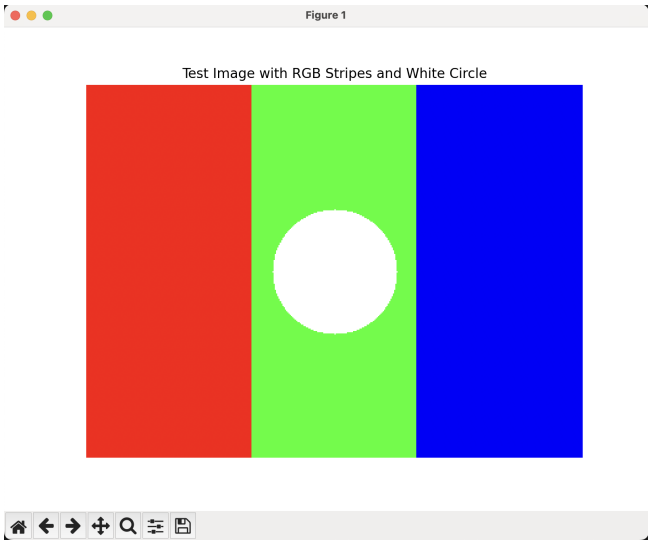


Gambar 12: Output script test audio

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



Gambar 13: Output script test image pada terminal



Gambar 14: Output script test image

4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

1. **Mengapa penting menggunakan environment terpisah untuk project multimedia?**

   Jawaban saya: Environment penting karena dapat memengaruhi kinerja teknis, pengujian, hingga pengalaman pengguna yang mana hal-hal tersebut sangat krusial untuk project multimedia.

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

Jawaban saya: Perbedaan utama antara conda, venv, dan uv adalah:

- *conda -> reliable tapi berat*
- *venv -> tidak terlalu berat tapi kecepatannya tergantung pip*
- *uv -> paling cepat di antara ketiga environment*

Saya memilih uv karena tidak berat untuk laptop saya dan juga tidak perlu menunggu lama untuk menginstallnya.

3. **Library mana yang paling sulit diinstall dan mengapa?**

   Jawaban saya: Selama penginstallan saya tidak menemukan kesulitan untuk menginstall library mana pun. Mungkin ada salah satu library yang memakan waktu lebih lama dari pada library-library lainnya. Library tersebut adalah Library General Purpose (numpy pandas jupyter). Proses penginstallan memang memakan waktu yang sedikit leboh lama tetapi tidak ada kesulitan dalam prosesnya.

4. **Bagaimana cara mengatasi masalah dependency conflict jika terjadi?**

   Jawaban saya: Seandainya terjadi dependency conflict, biasanya masalah itu timbul karena lebih dari 1 package memerlukan versi yang tidak kompatibel dari dependency yang sama. Jadi lebih baik buat environment baru, lalu tetapkan versi secara eksplisit dan instal semuanya sekaligus kemudian verifikasi.

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

   Jawaban saya:

   - *librosa -> untuk analisis musik dan audio: memuat file audio, menghitung spektrogram, mengekstrak fitur (tempo, pitch, MFCC), deteksi beat, dll.*
   - *soundfile -> untuk membaca dan menulis file suara.*
   - *scipy -> library komputasi ilmiah umum, dengan submodul scipy.signal untuk pemrosesan sinyal: penyaringan (filtering) audio, transformasi Fourier, interpolasi.*
   - *opencv-python (cv2) -> library computer vision untuk pemrosesan gambar/video secara real-time: transformasi gambar, deteksi wajah, pelacakan objek, deteksi tepi.*
   - *Pillow (PIL) -> untuk manipulasi gambar dasar: membuka, memotong, mengubah ukuran, memutar, mengonversi format (JPEG, PNG, dll.).*
   - *scikit-image (skimage) -> toolkit pemrosesan gambar tingkat lanjut berbasis NumPy/SciPy: segmentasi, denoising, morfologi, ekstraksi fitur.*
   - *matplotlib -> untuk plotting dan visualisasi: menampilkan gambar, membuat histogram, menggambar kurva data.*
   - *moviepy -> pengeditan video di Python: memotong, menggabungkan, menambah efek, menambahkan audio, membuat GIF.*
   - *ffmpeg (alat backend, tidak di-import langsung) -> program command-line yang kuat untuk mengonversi dan memproses video/audio.*
   - *numpy -> library inti untuk komputasi numerik di Python: operasi cepat pada array/matriks, aljabar linear, FFT.*
   - *pandas -> untuk analisis dan manipulasi data: membaca CSV/Excel, mengelola data tabular (baris kolom), grup, filter.*
   - *jupyter -> menyediakan environment Jupyter Notebook: pemrograman interaktif, menjalankan Python per cell, menampilkan grafik langsung di notebook.*

5 Export Environment untuk Reproduksi

Sebagai langkah terakhir, export environment Anda agar dapat direproduksi:

5.1 Untuk Conda

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

Kode 16: Export conda environment

5.2 Untuk venv/uv

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

Kode 17: Export pip requirements

Isi file requirements.txt:

```
anyio==4.10.0
appnope==0.1.4
argon2-cffi==25.1.0
argon2-cffi-bindings==25.1.0
arrow==1.3.0
asttokens==3.0.0
async-lru==2.0.5
attrs==25.3.0
audioop-lts==0.2.2
audioread==3.0.1
babel==2.17.0
beautifulsoup4==4.13.5
bleach==6.2.0
certifi==2025.8.3
cffi==1.17.1
charset-normalizer==3.4.3
comm==0.2.3
contourpy==1.3.3
cycler==0.12.1
debugpy==1.8.16
decorator==5.2.1
defusedxml==0.7.1
executing==2.2.0
fastjsonschema==2.21.2
fonttools==4.59.2
fqdn==1.5.1
h11==0.16.0
httpcore==1.0.9
httpx==0.28.1
idna==3.10
imageio==2.37.0
imageio-ffmpeg==0.6.0
ipykernel==6.30.1
ipython==9.4.0
ipython_pygments_lexers==1.1.1
ipywidgets==8.1.7
isoduration==20.11.0
jedi==0.19.2
Jinja2==3.1.6
joblib==1.5.2
json5==0.12.1
jsonpointer==3.0.0
jsonschema==4.25.1
```

```
jsonschema-specifications==2025.4.1
jupyter==1.1.1
jupyter-console==6.6.3
jupyter-events==0.12.0
jupyter-lsp==2.3.0
jupyter_client==8.6.3
jupyter_core==5.8.1
jupyter_server==2.17.0
jupyter_server_terminals==0.5.3
jupyterlab==4.4.6
jupyterlab_pygments==0.3.0
jupyterlab_server==2.27.3
jupyterlab_widgets==3.0.15
kiwisolver==1.4.9
lark==1.2.2
lazy_loader==0.4
librosa==0.11.0
llvmlite==0.44.0
MarkupSafe==3.0.2
matplotlib==3.10.5
matplotlib-inline==0.1.7
mistune==3.1.3
moviepy==2.2.1
msgpack==1.1.1
nbclient==0.10.2
nbconvert==7.16.6
nbformat==5.10.4
nest-asyncio==1.6.0
networkx==3.5
notebook==7.4.5
notebook_shim==0.2.4
numba==0.61.2
numpy==2.2.6
opencv-python==4.12.0.88
packaging==25.0
pandas==2.3.2
pandocfilters==1.5.1
parso==0.8.5
pexpect==4.9.0
pillow==11.3.0
platformdirs==4.4.0
pooch==1.8.2
proglog==0.1.12
prometheus_client==0.22.1
prompt_toolkit==3.0.52
psutil==7.0.0
ptyprocess==0.7.0
pure_eval==0.2.3
pycparser==2.22
Pygments==2.19.2
pyparsing==3.2.3
python-dateutil==2.9.0.post0
python-dotenv==1.1.1
python-json-logger==3.3.0
pytz==2025.2
PyYAML==6.0.2
pyzmq==27.0.2
referencing==0.36.2
requests==2.32.5
rfc3339-validator==0.1.4
rfc3986-validator==0.1.1
rfc3987-syntax==1.1.0
```

```
rpds-py==0.27.1
scikit-image==0.25.2
scikit-learn==1.7.1
scipy==1.16.1
Send2Trash==1.8.3
setuptools==80.9.0
six==1.17.0
sniffio==1.3.1
soundfile==0.13.1
soupsieve==2.8
soxr==0.5.0.post1
stack-data==0.6.3
standard-aifc==3.13.0
standard-chunk==3.13.0
standard-sunau==3.13.0
terminado==0.18.1
threadpoolctl==3.6.0
tifffile==2025.8.28
tinycss2==1.4.0
tornado==6.5.2
tqdm==4.67.1
traitlets==5.14.3
types-python-dateutil==2.9.0.20250822
typing_extensions==4.15.0
tzdata==2025.2
uri-template==1.3.0
urllib3==2.5.0
wcwidth==0.2.13
webcolors==24.11.1
webencodings==0.5.1
websocket-client==1.8.0
wheel==0.45.1
widgetsnbextension==4.0.14
```

Kode 18: Environment/Requirements file

6 Kesimpulan

Dari saya:

- Pengalaman setup Python environment untuk multimedia -> Awalnya saya kesulitan buat install environment karena saya belum terbiasa dengan command di macOS, tapi setelah tau command di macOS, saya jadi mudah untuk install environmentnya dan untuk commandnya kurang lebih sama seperti OS lainnya, hanya saja ada tambahan " python -m " sebelum menggunakan pip.
- Persiapan untuk project multimedia selanjutnya -> Mungkin saya akan coba untuk cari-cari gimana gambaran untuk audio, image, dan video processing. Setidaknya saya bisa tau workflownya.
- Saran untuk mahasiswa lain yang akan melakukan setup serupa -> Mungkin gunakan environment yang compatible dengan masing-masing device saja, jangan terlalu dipaksakan untuk pakai yang mungkin tidak memungkinkan untuk devicenya.

Environment itu penting adanya untuk project multimedia. Kenapa? Karena dengan adanya environment, performance teknisnya jadi lebih smooth dan pengujian project juga jadi lebih baik dan sesuai

7 Referensi

1. Pertanyaan seputar environment - ChatGPT
2. Membuat file Python sederhana untuk verifikasi instalasi - ChatGPT