



Nama: **Bagas Andreanto (122140017)**
Environment untuk Multimedia

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

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:

```
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]**
- Screenshot atau copy-paste output dari perintah verifikasi environment

```
C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>uv venv multimedia-uv
Using CPython 3.10.18
Creating virtual environment at: multimedia-uv
Activate with: multimedia-uv\Scripts\activate

C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>multimedia-uv\Scripts\activate

(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>which python
'which' is not recognized as an internal or external command,
operable program or batch file.

(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>which python
'which' is not recognized as an internal or external command,
operable program or batch file.

(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>where python
C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1\multimedia-uv\Scripts\python.exe
C:\Users\Pongo\AppData\Local\Programs\Python\Python312\python.exe
C:\Users\Pongo\AppData\Local\Microsoft\WindowsApps\python.exe
C:\msys64\mingw64\bin\python.exe
```

Gambar 1: Output verifikasi environment aktif di terminal

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

```

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 scikit_learn-1.7.1-cp312-cp312-win_amd64.whl (8.7 MB)
8.7/8.7 MB 2.4 MB/s 0:00:03
Downloading soxr-0.5.0.post1-cp312-abi3-win_amd64.whl (164 kB)
Downloading threadpoolctl-3.6.0-py3-none-any.whl (18 kB)
Downloading pycparser-2.22-py3-none-any.whl (117 kB)
Installing collected packages: threadpoolctl, pycparser, platformdirs, packaging, numpy, llvmlite, joblib, decorator, audioread, soxr, scipy,
pooch, numba, lazy_loader, cffi, soundfile, scikit-learn, librosa
Successfully installed audioread-3.0.1 cffi-1.17.1 decorator-5.2.1 joblib-1.5.1 lazy_loader-0.4 librosa-0.11.0 llvmlite-0.44.0 numba-0.61.2 nu
mpy-2.2.6 packaging-25.0 platformdirs-4.4.0 pooch-1.8.2 pycparser-2.22 scikit-learn-1.7.1 scipy-1.16.1 soundfile-0.13.1 soxr-0.5.0.post1 threa
dpoolctl-3.6.0

```

Gambar 2: Output sukses instalasi librosa, soundfile, scipy di terminal

```

Downloading contourpy-1.3.3-cp312-cp312-win_amd64.whl (226 kB)
Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.59.1-cp312-cp312-win_amd64.whl (2.3 MB)
2.3/2.3 MB 2.2 MB/s 0:00:01
Downloading imageio-2.37.0-py3-none-any.whl (315 kB)
Downloading kiwisolver-1.4.9-cp312-cp312-win_amd64.whl (73 kB)
Downloading networkx-3.5-py3-none-any.whl (2.0 MB)
2.0/2.0 MB 2.1 MB/s 0:00:00
Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Downloading tifffile-2025.6.11-py3-none-any.whl (230 kB)
Installing collected packages: tifffile, six, pillow, opencv-python, networkx, kiwisolver, fonttools, cycler, contourpy, python-dateutil, imag
eio, scikit-image, matplotlib
Successfully installed contourpy-1.3.3 cycler-0.12.1 fonttools-4.59.1 imageio-2.37.0 kiwisolver-1.4.9 matplotlib-3.10.5 networkx-3.5 opencv-py
thon-4.12.0.88 pillow-11.3.0 python-dateutil-2.9.0.post0 scikit-image-0.25.2 six-1.17.0 tifffile-2025.6.11

```

Gambar 3: Output sukses instalasi opencv, pillow, scikit-image, matplotlib di terminal

```

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\pongo\appdata\local\programs\python\python312\lib\site-packages (from moviepy)
(11.3.0)
Collecting tqdm (from proglog<=1.0.0->moviepy)
  Downloading tqdm-4.67.1-py3-none-any.whl.metadata (57 kB)
Requirement already satisfied: colorama in c:\users\pongo\appdata\local\programs\python\python312\lib\site-packages (from tqdm->proglog<=1.0.0->moviepy) (0.4.6)
Downloading moviepy-2.2.1-py3-none-any.whl (129 kB)
Downloading proglog-0.1.12-py3-none-any.whl (6.3 kB)
Downloading imageio_ffmpeg-0.6.0-py3-none-win_amd64.whl (31.2 MB)
31.2/31.2 MB 1.1 MB/s 0:00:28
Downloading python_dotenv-1.1.1-py3-none-any.whl (20 kB)
Downloading tqdm-4.67.1-py3-none-any.whl (78 kB)
Installing collected packages: tqdm, python-dotenv, imageio_ffmpeg, proglog, moviepy
Successfully installed imageio_ffmpeg-0.6.0 moviepy-2.2.1 proglog-0.1.12 python-dotenv-1.1.1 tqdm-4.67.1

```

Gambar 4: Output sukses instalasi moviepy di terminal

Gambar 5: Output sukses instalasi numpy, pandas, jupyter di terminal

- Daftar library yang berhasil diinstall dengan versinya
 - jupyter 1.1.1
 - librosa 0.11.0
 - matplotlib 3.10.5
 - matplotlib-inline 0.1.7
 - moviepy 2.2.1
 - numpy 2.2.6
 - opencv-python 4.12.0.88
 - pandas 2.3.2
 - pillow 11.3.0
 - scikit-image 0.25.2
 - scipy 1.16.1
 - soundfile 0.13.1

3.4 Bagian 3: Verifikasi Instalasi

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

Jalankan script dan dokumentasikan hasilnya:

```
1 # Verifikasi-Instalasi.py
2 print("=== Verifikasi Library ===")
3
4 try:
5     import librosa
6     print(f"librosa sudah terinstal, versi: {librosa.__version__}")
7 except ImportError:
8     print("librosa BELUM terinstal")
9
10 try:
11     import soundfile
12     print(f"soundfile sudah terinstal, versi: {soundfile.__version__}")
13 except ImportError:
14     print("soundfile BELUM terinstal")
15
16 try:
17     import scipy
18     print(f"scipy sudah terinstal, versi: {scipy.__version__}")
19 except ImportError:
```

```
20     print("scipy BELUM terinstal")
21
22 try:
23     import cv2
24     print(f"opencv-python sudah terinstal, versi: {cv2.__version__}")
25 except ImportError:
26     print("opencv-python BELUM terinstal")
27
28 try:
29     import PIL
30     print(f"Pillow sudah terinstal, versi: {PIL.__version__}")
31 except ImportError:
32     print("Pillow BELUM terinstal")
33
34 try:
35     import skimage
36     print(f"scikit-image sudah terinstal, versi: {skimage.__version__}")
37 except ImportError:
38     print("scikit-image BELUM terinstal")
39
40 try:
41     import matplotlib
42     print(f"matplotlib sudah terinstal, versi: {matplotlib.__version__}")
43 except ImportError:
44     print("matplotlib BELUM terinstal")
45
46 try:
47     import moviepy
48     print(f"moviepy sudah terinstal, versi: {moviepy.__version__}")
49 except ImportError:
50     print("moviepy BELUM terinstal")
51
52 try:
53     import numpy
54     print(f"numpy sudah terinstal, versi: {numpy.__version__}")
55 except ImportError:
56     print("numpy BELUM terinstal")
57
58 try:
59     import pandas
60     print(f"pandas sudah terinstal, versi: {pandas.__version__}")
61 except ImportError:
62     print("pandas BELUM terinstal")
63
64 try:
65     import notebook
66     print(f"jupyter-notebook sudah terinstal, versi: {notebook.__version__}")
67 except ImportError:
68     print("jupyter-notebook BELUM terinstal")
69
70 print("=== Verifikasi selesai ===")
```

Kode 8: Kode script verifikasi instalasi di file test_multimedia.py

```
(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>python test_multimedia.py
=== Verifikasi Library ===
librosa sudah terinstal, versi: 0.11.0
soundfile sudah terinstal, versi: 0.13.1
scipy sudah terinstal, versi: 1.15.3
opencv-python sudah terinstal, versi: 4.12.0
Pillow sudah terinstal, versi: 11.3.0
scikit-image sudah terinstal, versi: 0.25.2
matplotlib sudah terinstal, versi: 3.10.5
moviepy sudah terinstal, versi: 2.1.2
numpy sudah terinstal, versi: 2.2.6
pandas sudah terinstal, versi: 2.3.2
jupyter-notebook sudah terinstal, versi: 7.4.5
=== Verifikasi selesai ===
```

Gambar 6: Hasil Output dari script test_multimedia.py

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 9: 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 10: Test image processing sederhana

Dokumentasikan hasil eksekusi:

- Screenshot output dari kedua script di atas

```

(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>python test_audio.py
Generated 2s sine wave at 440Hz
Sample rate: 44100Hz
Total samples: 88200

(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>

```

Gambar 7: Output dari script pertama

```

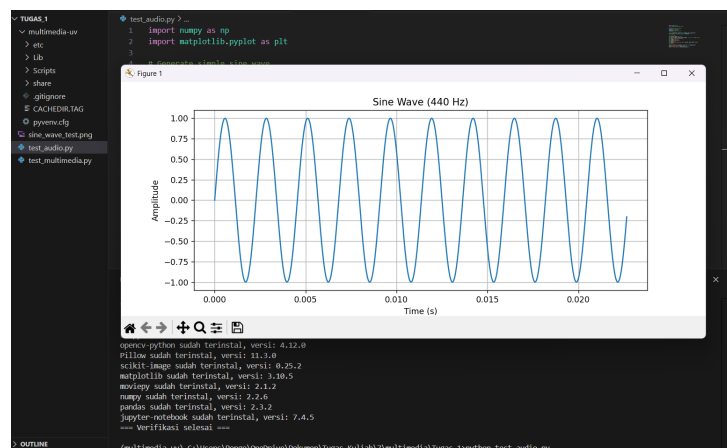
(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>python test_image.py
Created test image: 400x300 pixels
Image shape: (300, 400, 3)
Image dtype: uint8

(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>

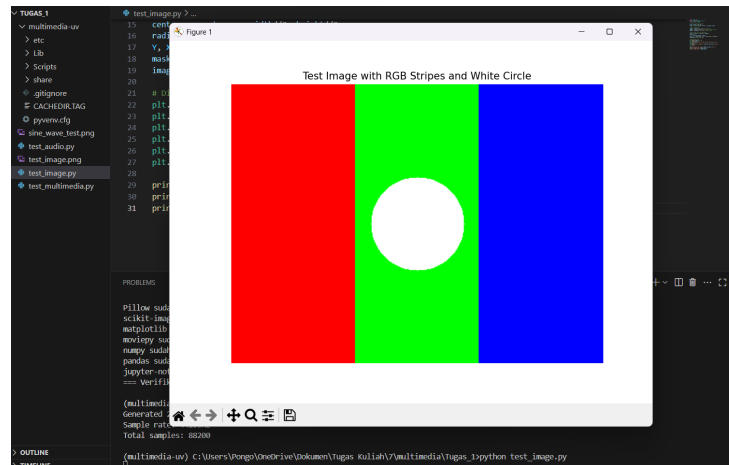
```

Gambar 8: Output dari script kedua

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



Gambar 9: Hasil gambar sine_wave_test.png



Gambar 10: Hasil gambar 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  === Verifikasi Library ===
2  librosa sudah terinstal, versi: 0.11.0
3  soundfile sudah terinstal, versi: 0.13.1
4  scipy sudah terinstal, versi: 1.15.3
5  opencv-python sudah terinstal, versi: 4.12.0
6  Pillow sudah terinstal, versi: 11.3.0
7  scikit-image sudah terinstal, versi: 0.25.2
8  matplotlib sudah terinstal, versi: 3.10.5
9  moviepy sudah terinstal, versi: 2.1.2
10 numpy sudah terinstal, versi: 2.2.6
11 pandas sudah terinstal, versi: 2.3.2
12 jupyter-notebook sudah terinstal, versi: 7.4.5
13 === Verifikasi selesai ===

```

Kode 11: Output verifikasi instalasi

4.2 Screenshot Hasil Test

Sisipkan screenshot atau gambar hasil dari:

- Terminal/command prompt yang menunjukkan environment aktif

```

C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>uv venv multimedia-uv
Using CPython 3.10.18
Creating virtual environment at: multimedia-uv
Activate with: multimedia-uv\Scripts\activate

C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>multimedia-uv\Scripts\activate

(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>which python
'which' is not recognized as an internal or external command,
operable program or batch file.

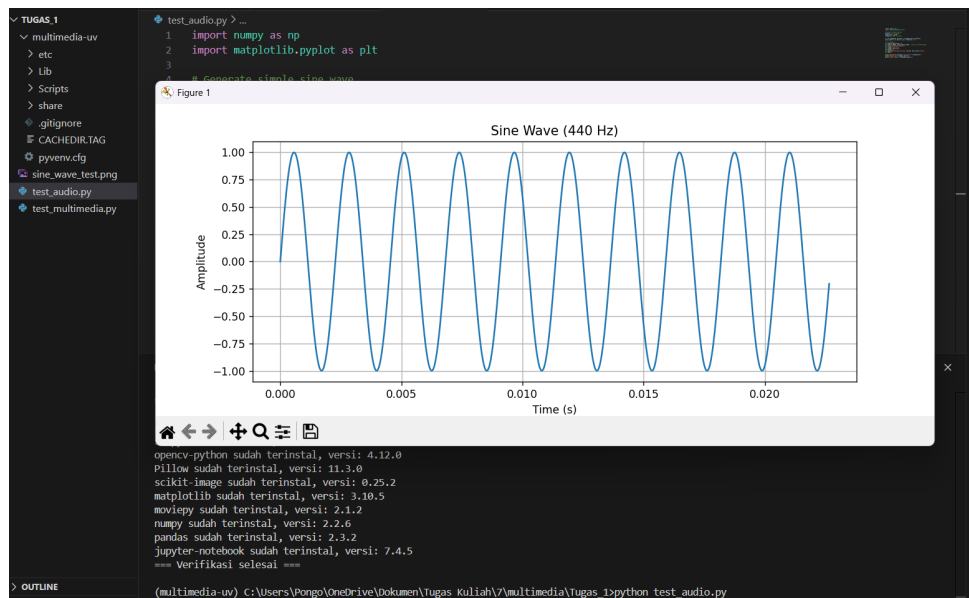
(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>which python
'which' is not recognized as an internal or external command,
operable program or batch file.

(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>where python
C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1\multimedia-uv\Scripts\python.exe
C:\Users\Pongo\AppData\Local\Programs\Python\Python312\python.exe
C:\Users\Pongo\AppData\Local\Microsoft\WindowsApps\python.exe
C:\msys64\mingw64\bin\python.exe

```

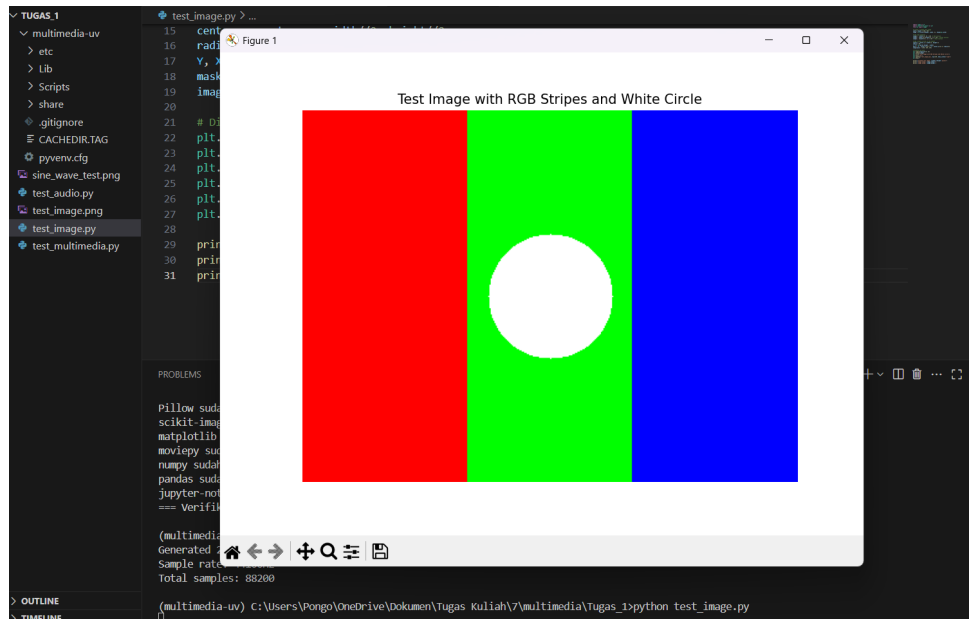
Gambar 11: Tampilan terminal yang menunjukkan environment aktif

- Output dari script test audio (sine wave plot)



Gambar 12: Hasil plot sine wave dari script test audio

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



Gambar 13: Hasil gambar RGB stripes dengan lingkaran dari script test image

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, menggunakan environment terpisah untuk setiap project multimedia itu penting karena setiap project biasanya memiliki kebutuhan library dan versi yang berbeda-beda. Jadi dengan environment terpisah, maka kita dapat menghindari terjadinya konflik antar library yang dapat menyebabkan error.

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

Berdasarkan informasi yang saya dapatkan di pertemuan pertama kemarin, ketiga tools manajemen environment tersebut memiliki perbedaan utama pada kecepatan, ukuran, serta karakteristik dependency managementnya.

Untuk conda dan venv, memiliki kekurangan dari segi ukuran serta proses instalasinya yang cukup lama. Sedangkan uv memiliki keunggulan dari segi kecepatan dan ukuran yang lebih kecil, meskipun memiliki kekurangan yaitu harus melakukan instalasi ulang setiap berpindah folder project.

3. Library mana yang paling sulit diinstall dan mengapa?

Sebenarnya tidak ada library yang sulit untuk diinstall, hanya saja saya kurang teliti dalam proses instalasi sehingga library terinstall secara global dan bukan di environment aktif yang sebelumnya sudah di buat.

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

Melakukan upgrade untuk semua library yang bermasalah, maka uv akan otomatis menginstall versi yang kompatibel. Kemudian, bisa juga menggunakan uv pip check untuk memeriksa, library mana saja yang bertabrakan versinya, kemudian bisa di hapus ataupun di upgrade.

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

- **jupyter:** Menyediakan notebook interaktif untuk menulis kode, teks, grafik, dan hasil eksekusi
- **librosa:** Untuk analisis audio/musik, seperti mengekstrak file audio
- **matplotlib:** Untuk membuat grafik, plot, dan chart
- **moviepy:** Untuk mengedit dan memproses video
- **numpy:** Untuk komputasi numerik dengan array/matriks
- **opencv-python:** Library computer vision untuk membaca, memproses, dan menganalisis gambar atau video.
- **pandas:** Untuk analisis dan manipulasi data, terutama data berbentuk tabel.
- **pillow:** Untuk manipulasi gambar, seperti resize, crop, filter, dan konversi format.
- **scikit-image:** Untuk pengolahan citra, termasuk filter dan transformasi gambar.
- **scipy:** Menyediakan algoritma untuk optimisasi, aljabar linear, FFT, statistik, dan lain-lain.
- **soundfile:** Untuk membaca dan menulis file audio dengan format seperti WAV atau FLAC.

4.4 Troubleshooting

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

- **Masalah 1:** Kesalahan penggunaan `which python` pada saat verifikasi environment aktif dengan `uv`.

Solusi: Saya mencari informasi melalui chat GPT, dan menemukan bahwa command `which python` hanya dapat digunakan pada macbook/linux, sedangkan pada windows harus menggunakan command `where python`.

- **Masalah 2:** Menginstall library secara global, bukan di environment aktif.

Solusi: Saya mencari informasi melalui chat GPT, dan menemukan bahwa saya kurang teliti dalam proses instalasi library. Yang mana ketika saya melakukan instalasi semua library sebelumnya, saya langsung menuliskan command `pip install` tanpa memastikan bahwa environment yang saya buat sebelumnya sudah aktif. Sehingga semua library yang di install masih berada di global dan bukan berada di environment yang sudah saya buat.

```
(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>python -m pip --version
C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1\multimedia-uv\Scripts\python.exe: No module named pip

(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>python -m ensurepip --upgrade
Looking in links: c:\Users\Pongo\AppData\Local\Temp\tmpwutt01_i
Processing c:\users\pongo\appdata\local\temp\tmpwutt01_i\setuptools-65.5.0-py3-none-any.whl
Processing c:\users\pongo\appdata\local\temp\tmpwutt01_i\pip-23.0.1-py3-none-any.whl
Installing collected packages: setuptools, pip
Successfully installed pip-23.0.1 setuptools-65.5.0

(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>python Verifikasi-Instalasi.py
librosa BELUM terinstal

(multimedia-uv) C:\Users\Pongo\OneDrive\Dokumen\Tugas Kuliah\7\multimedia\Tugas_1>python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\pongo\onedrive\dokumen\tugas kuliah\7\multimedia\tugas_1\multimedia-uv\lib\site-packages (23.0.1)
```

Gambar 14: Output solusi masalah instalasi library di terminal

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 12: Export conda environment

5.2 Untuk venv/uv

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

Kode 13: Export pip requirements

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

```
1 anyio==4.10.0
2 argon2-cffi==25.1.0
3 argon2-cffi-bindings==25.1.0
4 arrow==1.3.0
5 asttokens==3.0.0
6 async-lru==2.0.5
7 attrs==25.3.0
8 audioread==3.0.1
9 babel==2.17.0
10 beautifulsoup4==4.13.5
11 bleach==6.2.0
12 certifi==2025.8.3
13 cffi==1.17.1
14 charset-normalizer==3.4.3
15 colorama==0.4.6
16 comm==0.2.3
17 contourpy==1.3.2
18 cycler==0.12.1
19 debugpy==1.8.16
20 decorator==5.2.1
21 defusedxml==0.7.1
22 exceptiongroup==1.3.0
23 executing==2.2.0
24 fastjsonschema==2.21.2
25 fonttools==4.59.1
26 fqdn==1.5.1
27 h11==0.16.0
28 httpcore==1.0.9
29 httpx==0.28.1
30 idna==3.10
31 imageio==2.37.0
32 imageio-ffmpeg==0.6.0
33 ipykernel==6.30.1
34 ipython==8.37.0
35 ipywidgets==8.1.7
36 isoduration==20.11.0
37 jedi==0.19.2
38 Jinja2==3.1.6
39 joblib==1.5.2
40 json5==0.12.1
41 jsonpointer==3.0.0
42 jsonschema==4.25.1
43 jsonschema-specifications==2025.4.1
44 jupyter==1.1.1
45 jupyter-console==6.6.3
46 jupyter-events==0.12.0
47 jupyter-lsp==2.2.6
48 jupyter_client==8.6.3
```

```
49 jupyter_core==5.8.1
50 jupyter_server==2.17.0
51 jupyter_server_terminals==0.5.3
52 jupyterlab==4.4.6
53 jupyterlab_pygments==0.3.0
54 jupyterlab_server==2.27.3
55 jupyterlab_widgets==3.0.15
56 kiwisolver==1.4.9
57 lark==1.2.2
58 lazy_loader==0.4
59 librosa==0.11.0
60 llvmlite==0.44.0
61 MarkupSafe==3.0.2
62 matplotlib==3.10.5
63 matplotlib-inline==0.1.7
64 mistune==3.1.3
65 moviepy==2.2.1
66 msgpack==1.1.1
67 nbclient==0.10.2
68 nbconvert==7.16.6
69 nbformat==5.10.4
70 nest-asyncio==1.6.0
71 networkx==3.4.2
72 notebook==7.4.5
73 notebook_shim==0.2.4
74 numba==0.61.2
75 numpy==2.2.6
76 opencv-python==4.12.0.88
77 overrides==7.7.0
78 packaging==25.0
79 pandas==2.3.2
80 pandocfilters==1.5.1
81 parso==0.8.5
82 pillow==11.3.0
83 platformdirs==4.4.0
84 pooch==1.8.2
85 proglog==0.1.12
86 prometheus_client==0.22.1
87 prompt_toolkit==3.0.51
88 psutil==7.0.0
89 pure_eval==0.2.3
90 pycparser==2.22
91 Pygments==2.19.2
92 pyparsing==3.2.3
93 python-dateutil==2.9.0.post0
94 python-dotenv==1.1.1
95 python-json-logger==3.3.0
96 pytz==2025.2
97 pywin32==311
98 pywinpty==3.0.0
99 PyYAML==6.0.2
100 pyzmq==27.0.2
101 referencing==0.36.2
102 requests==2.32.5
103 rfc3339-validator==0.1.4
104 rfc3986-validator==0.1.1
105 rfc3987-syntax==1.1.0
106 rpds-py==0.27.1
107 scikit-image==0.25.2
108 scikit-learn==1.7.1
109 scipy==1.15.3
110 Send2Trash==1.8.3
```

```
111 six==1.17.0
112 sniffio==1.3.1
113 soundfile==0.13.1
114 soupsieve==2.7
115 soxr==0.5.0.post1
116 stack-data==0.6.3
117 terminado==0.18.1
118 threadpoolctl==3.6.0
119 tiffiff==2025.5.10
120 tinycss2==1.4.0
121 tomli==2.2.1
122 tornado==6.5.2
123 tqdm==4.67.1
124 traitlets==5.14.3
125 types-python-dateutil==2.9.0.20250822
126 typing_extensions==4.15.0
127 tzdata==2025.2
128 uri-template==1.3.0
129 urllib3==2.5.0
130 wcwidth==0.2.13
131 webcolors==24.11.1
132 webencodings==0.5.1
133 websocket-client==1.8.0
134 widgetsnbextension==4.0.14
```

Kode 14: 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 yang saya dapatkan selama proses setup python environment ini sebenarnya cukup simple, hanya saja saya kurang fokus dan teliti dalam beberapa langkah, sehingga menyebabkan beberapa masalah ketika proses instalasinya.

Untuk persiapan project selanjutnya, mungkin saya harus lebih berkonsentrasi ketika mengikuti step by step nya.

Saran dari saya yaitu, untuk dapat lebih cepat dikerjakan, karena menurut saya tugas ini tidak sesimple kelihatannya

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

References

- [1] OpenAI. *ChatGPT - Conversational AI*. Tersedia di: <https://chatgpt.com/share/68b03039-5e04-8006-96cd-cf3ad0606c09> (diakses 28 Agustus 2025).

References