

INSTITUT TEKNOLOGI SUMATERA

FAKULTAS INFORMATIKA

Nama: **Naufal Haris Nurkhoirulloh (122140040)** 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 \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

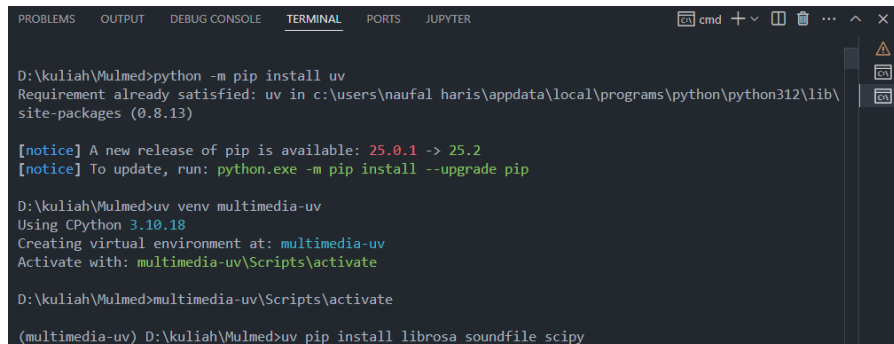
3.2.3 Opsi 3: Menggunakan uv (Modern dan cepat)

```
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



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER
D:\kuliah\Mulmed>python -m pip install uv
Requirement already satisfied: uv in c:\users\naufal haris\appdata\local\programs\python\python312\lib\
site-packages (0.8.13)

[notice] A new release of pip is available: 25.0.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip

D:\kuliah\Mulmed>uv venv multimedia-uv
Using CPython 3.10.18
Creating virtual environment at: multimedia-uv
Activate with: multimedia-uv\Scripts\activate

D:\kuliah\Mulmed>multimedia-uv\Scripts\activate
(multimedia-uv) D:\kuliah\Mulmed>uv pip install librosa soundfile scipy
```

Gambar 1. Output verifikasi 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

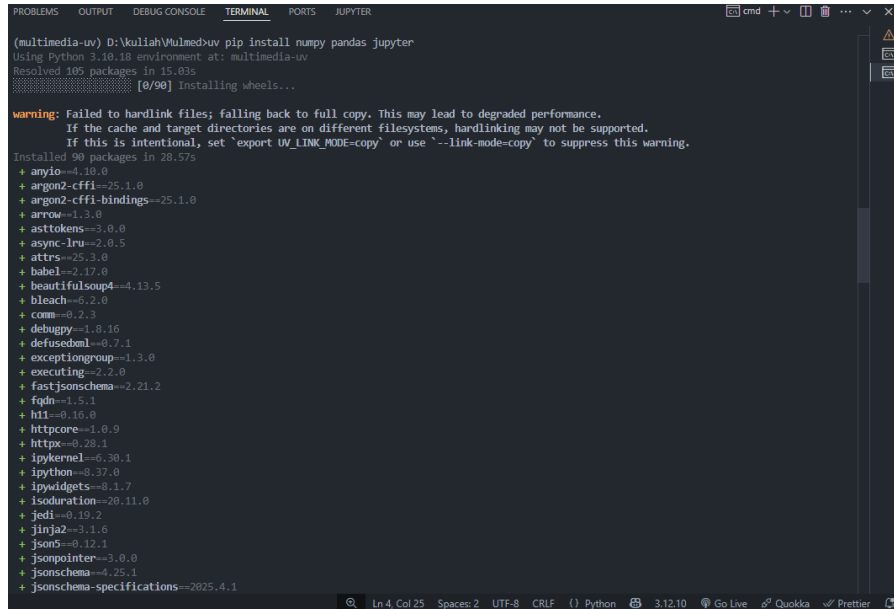
3.3.4 Library General Purpose

```
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: **uv pip install**
- Screenshot proses instalasi atau output sukses:

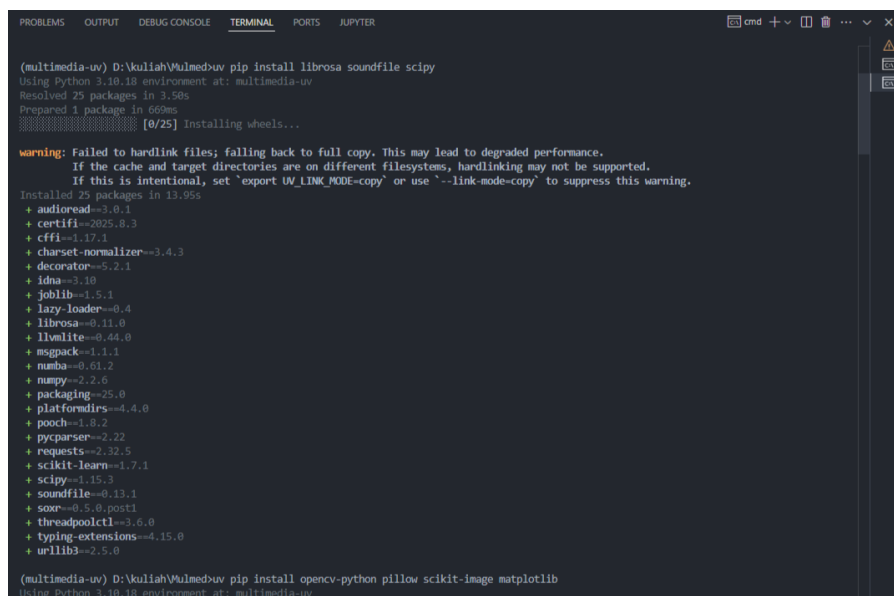


```
(multimedia-uv) D:\Kuliah\VuImed>uv pip install numpy pandas jupyter
Using Python 3.10.18 environment at: multimedia-uv
Resolved 105 packages in 15.03s
[0/90] Installing wheels...

warning: Failed to hardlink files; falling back to full copy. This may lead to degraded performance.
If the cache and target directories are on different filesystems, hardlinking may not be supported.
If this is intentional, set 'export UV_LINK_MODE=copy' or use '--link-mode=copy' to suppress this warning.

Installed 90 packages in 28.57s
+ anyio==4.10.0
+ 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
+ babel==2.17.0
+ beautifulsoup4==4.13.5
+ bleach==6.2.0
+ comm==0.2.3
+ debugpy==1.8.16
+ defusedxml==0.7.1
+ exceptiongroup==1.3.0
+ executing==2.2.0
+ fastjsonschema==2.21.2
+ fqdn==1.5.1
+ h11==0.16.0
+ httpcore==1.0.9
+ httpx==0.28.1
+ ipykernel==6.30.1
+ ipython==8.37.0
+ ipywidgets==8.1.7
+ isoduration==20.11.0
+ jedi==0.19.2
+ Jinja2==3.1.6
+ json5==0.12.1
+ jsonpointer==3.0.0
+ jsonschema==4.25.1
+ jsonschema-specifications==2025.4.1
```

Gambar 2. Instalasi library umum



```
(multimedia-uv) D:\Kuliah\VuImed>uv pip install librosa soundfile scipy
Using Python 3.10.18 environment at: multimedia-uv
Resolved 25 packages in 3.50s
Prepared 1 package in 669ms
[0/25] Installing wheels...

warning: Failed to hardlink files; falling back to full copy. This may lead to degraded performance.
If the cache and target directories are on different filesystems, hardlinking may not be supported.
If this is intentional, set 'export UV_LINK_MODE=copy' or use '--link-mode=copy' to suppress this warning.

Installed 25 packages in 13.95s
+ audioread==3.0.1
+ certifi==2025.8.3
+ cffi==1.17.1
+ charset-normalizer==3.4.3
+ decorator==5.2.1
+ idna==3.10
+ joblib==1.5.1
+ lazy-loader==0.4
+ librosa==0.11.0
+ llvmlite==0.44.0
+ msgpack==1.1.1
+ numba==0.61.2
+ numpy==2.2.6
+ packaging==25.0
+ platformdirs==4.4.0
+ pooch==1.0.2
+ pycparser==2.22
+ requests==2.32.5
+ scikit-learn==1.7.1
+ scipy==1.15.3
+ soundfile==0.13.1
+ soxr==0.5.0.post1
+ threadpoolctl==3.6.0
+ typing-extensions==4.15.0
+ urllib3==2.5.0

(multimedia-uv) D:\Kuliah\VuImed>uv pip install opencv-python pillow scikit-image matplotlib
Using Python 3.10.18 environment at: multimedia-uv
```

Gambar 3. Instalasi library audio

```

(multimedia-uv) D:\kuliah\Mulmed>uv pip install opencv-python pillow scikit-image matplotlib
Using Python 3.10.18 environment at: multimedia-uv
Resolved 18 packages in 1.85s
[0/14] Installing wheels...

warning: Failed to hardlink files; falling back to full copy. This may lead to degraded performance.
If the cache and target directories are on different filesystems, hardlinking may not be supported.
If this is intentional, set `export UV_LINK_MODE=copy` or use `--link-mode=copy` to suppress this warning.

Installed 14 packages in 5.65s
+ contourpy==1.3.2
+ cycler==0.12.1
+ fonttools==4.59.1
+ imageio==2.37.0
+ kiwisolver==1.4.9
+ matplotlib==3.10.5
+ networkx==3.4.2
+ opencv-python==4.12.0.88
+ pillow==11.3.0
+ pyarsing==3.2.3
+ python-dateutil==2.9.0.post0
+ scikit-image==0.25.2
+ six==1.17.0
+ tifffile==2025.5.10

```

Gambar 4. Instalasi library image processing

```

(multimedia-uv) D:\kuliah\Mulmed>uv pip install moviepy
Using Python 3.10.18 environment at: multimedia-uv
Resolved 10 packages in 1.59s
[0/6] Installing wheels...

warning: Failed to hardlink files; falling back to full copy. This may lead to degraded performance.
If the cache and target directories are on different filesystems, hardlinking may not be supported.
If this is intentional, set `export UV_LINK_MODE=copy` or use `--link-mode=copy` to suppress this warning.

Installed 6 packages in 1.47s
+ 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

```

Gambar 5. Instalasi library video processing

- Saya menggunakan perintah `uv pip list` untuk melihat library yang sudah terinstall, berikut adalah daftar library yang berhasil diinstall:

audioread	3.0.1
imageio	2.37.0
imageio-ffmpeg	0.6.0
ipykernel	6.30.1
ipython	8.37.0
ipywidgets	8.1.7
isoduration	20.11.0
jupyter	1.1.1
jupyter-client	8.6.3
jupyter-console	6.6.3
jupyter-core	5.8.1
jupyter-events	0.12.0
jupyter-lsp	2.2.6
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
librosa	0.11.0
matplotlib	3.10.5
matplotlib-inline	0.1.7
moviepy	2.2.1
notebook	7.4.5
notebook-shim	0.2.4
numba	0.61.2
numpy	2.2.6
opencv-python	4.12.0.88
pandas	2.3.2
pillow	11.3.0
proglog	0.1.12
scikit-image	0.25.2
scikit-learn	1.7.1
scipy	1.15.3
soundfile	0.13.1
soxr	0.5.0.post1
tifffile	2025.5.10
tqdm	4.67.1
traitlets	5.14.3

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:

```

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 # Hz
7 frequency = 440 # Ad 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(block=False)
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)}")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

(multimedia-uv) D:\Kuliah\VuMed\python testAudio.py

(multimedia-uv) D:\Kuliah\VuMed\python testAudio.py
Generated 2's sine wave at 440Hz.
Generated 2's sine wave at 440Hz.
Sample Rate: 44100Hz
Total Samples: 88200

Gambar 6. Output script audio

```

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 the image
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")

```

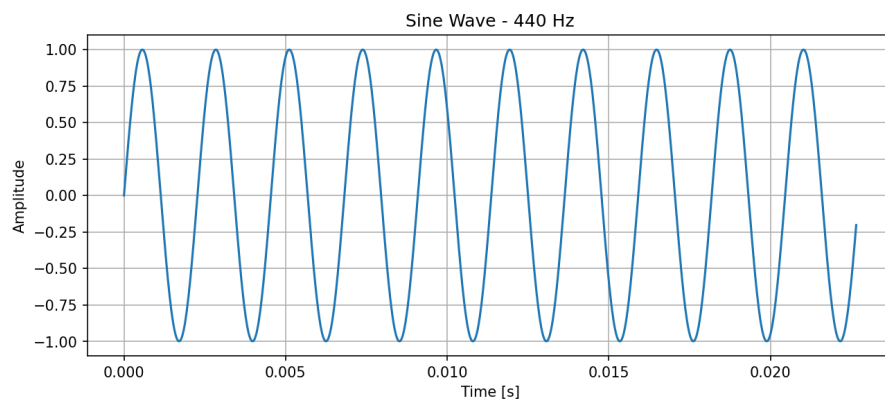
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

(multimedia-uv) D:\Kuliah\VuMed\python testImage.py

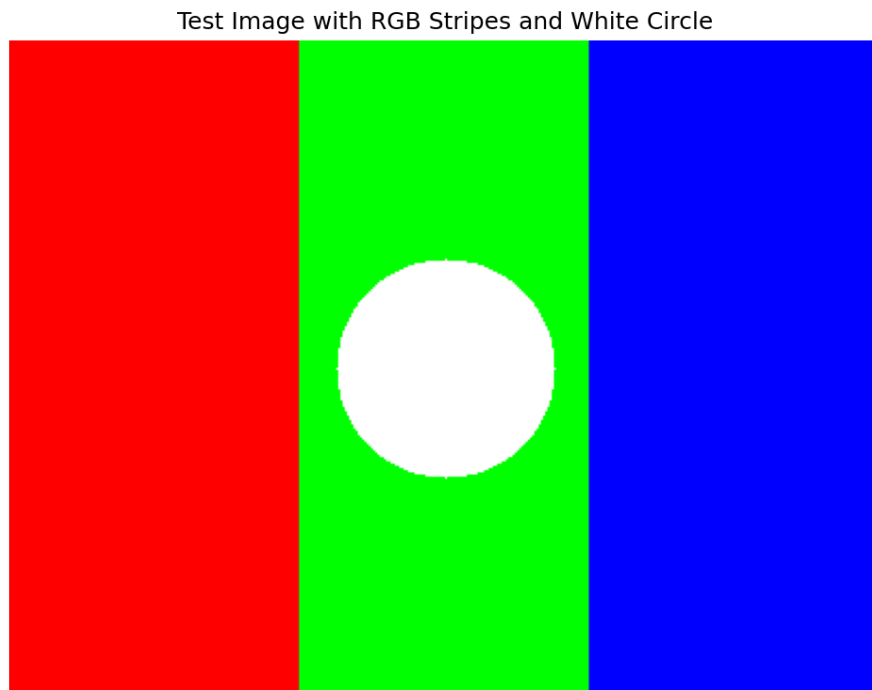
(multimedia-uv) D:\Kuliah\VuMed\python testImage.py
Created test image: 400x300 pixels
Created test image: 400x300 pixels
Image shape: (300, 400, 3)
Image dtype: uint8

Gambar 7. Output script image

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



Gambar 8. Gambar audio processing : sine_wave_test.png



Gambar 9. Gambar image processing : test_image.png

- Error message jika ada dan cara mengatasinya

Output perintah `print(...)` tidak muncul di terminal karena program berhenti di `plt.show()` pada matplotlib membuka jendela baru untuk menampilkan plot. Jadi saya menambahkan `plt.show(block=False)` pada kedua code agar hasil output muncul di terminal.

4 Bagian Laporan

4.1 Output Verifikasi Instalasi

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

```
1 # Verifikasi instalasi test audio processing
2 (multimedia-uv) D:\kuliah\Mulmed>python testAudio.py
3 Generated 2's sine wave at 440Hz.
4 Generated 2's sine wave at 440Hz.
5 Sample Rate: 44100Hz
6 Total Samples: 88200
7
8 # Verifikasi instalasi test image processing
9 (multimedia-uv) D:\kuliah\Mulmed>python testImage.py
10 Created test image: 400x300 pixels
11 Created test image: 400x300 pixels
12 Image shape: (300, 400, 3)
13 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

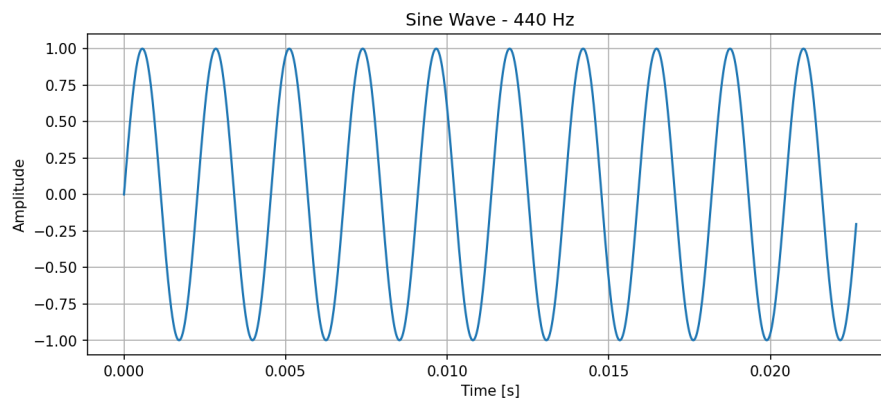
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

(multimedia-uv) D:\kuliah\Mulmed>deactivate
D:\kuliah\Mulmed>multimedia-uv\scripts\activate

(multimedia-uv) D:\kuliah\Mulmed>
(multimedia-uv) D:\kuliah\Mulmed>
(multimedia-uv) D:\kuliah\Mulmed>
(multimedia-uv) D:\kuliah\Mulmed>sudah aktif
```

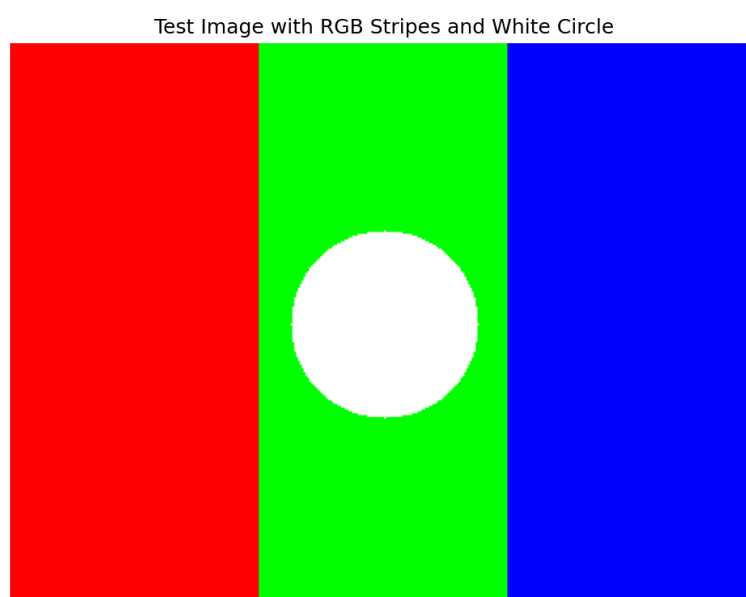
Gambar 10. Mengaktifkan enviroentment uv

- Output dari script test audio (sine wave plot)



Gambar 11. Output sine wave test

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



Gambar 12. Output RGB stripes dengan circle

4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

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

Agar lebih ringan dan bersih, tidak merusak project lain saat melakukan eksperimen, dan aman dari konflik antar library, serta memudahkan pengelolaan dependensi spesifik untuk setiap project.

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

Conda adalah package manager yang dapat mengelola paket dan environment, venv adalah modul bawaan Python untuk membuat virtual environment, dan uv adalah tool yang lebih ringan dan cepat untuk membuat virtual environment. Saya memilih uv karena lebih sederhana dan cepat dalam membuat environment baru.

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

Untuk saat ini tidak ada library yang sulit diinstall karena semua berjalan lancar.

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

Dengan mengecek dependency yang konflik dan mencari versi yang sesuai bisa di upgrade atau downgrade agar kompatibel.

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

- (a) **librosa** : Digunakan untuk analisis dan pemrosesan sinyal audio (misalnya ekstraksi fitur MFCC, spektrogram, tempo).
- (b) **soundfile** : Membaca dan menulis file audio dalam berbagai format (seperti WAV, FLAC, OGG).
- (c) **scipy** : Library ilmiah yang menyediakan fungsi matematika, optimisasi, sinyal, dan pemrosesan data.
- (d) **opencv-python** : Library untuk pengolahan citra dan computer vision (misalnya deteksi objek, transformasi gambar, filtering).
- (e) **pillow** : Library untuk manipulasi gambar (membuka, mengedit, menyimpan file gambar dengan berbagai format).
- (f) **scikit-image** : Digunakan untuk pemrosesan citra berbasis sains (misalnya segmentasi, denoising, fitur gambar).
- (g) **matplotlib** : Library visualisasi untuk membuat plot, grafik, dan menampilkan gambar.
- (h) **moviepy** : Library untuk pengolahan video (editing, efek, komposisi, dan konversi format).
- (i) **numpy** : Library fundamental untuk komputasi numerik (array multidimensi, operasi matematis).
- (j) **pandas** : Digunakan untuk analisis data, menyediakan struktur data yang efisien.
- (k) **jupyter** : Platform interaktif untuk menjalankan kode Python dalam bentuk notebook yang mendukung teks, visualisasi, dan eksekusi kode.

4.4 Troubleshooting

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

- **Masalah 1:** Output print tidak muncul di terminal

Solusi: Menambahkan `plt.show(block=False)` pada kedua code agar hasil output muncul di terminal. Saya menganalisis masalahnya dengan bantuan chatGPT

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 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==1.5.1
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-client==8.6.3
46 jupyter-console==6.6.3
47 jupyter-core==5.8.1
48 jupyter-events==0.12.0
49 jupyter-lsp==2.2.6
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==0.27.0
107 scikit-image==0.25.2
108 scikit-learn==1.7.1
109 scipy==1.15.3
110 send2trash==1.8.3
111 setuptools==80.9.0
112 six==1.17.0
113 sniffio==1.3.1
114 soundfile==0.13.1
115 soupsieve==2.7
116 soxr==0.5.0.post1
117 stack-data==0.6.3
118 terminado==0.18.1
119 threadpoolctl==3.6.0
120 tifffile==2025.5.10
121 tinycss2==1.4.0
122 tomli==2.2.1
123 tornado==6.5.2
124 tqdm==4.67.1
125 traitlets==5.14.3
126 types-python-dateutil==2.9.0.20250822
127 typing-extensions==4.15.0
128 tzdata==2025.2
129 uri-template==1.3.0
130 urllib3==2.5.0
131 wcwidth==0.2.13
132 webcolors==24.11.1
133 webencodings==0.5.1
134 websocket-client==1.8.0
135 widgetsnbextension==4.0.14
```

Kode 13. Environment/Requirements file

6 Kesimpulan

Tuliskan kesimpulan Anda mengenai:

- Pengalaman setup Python environment untuk multimedia

Awalnya saya menggunakan terminal powershell dan ada beberapa kendala dimana harus run as administrator baru bisa aktivasi environment. Kemudian saya pindah menggunakan terminal vscode dan semua berjalan lancar. Proses instalasi library juga berjalan lancar tanpa ada error namun saya pakai uv pip install baru bisa.
- Persiapan untuk project multimedia selanjutnya

Saya sudah menyiapkan folder untuk project kelas multimedia dengan environment uv yang sudah terinstall library multimedia.
- Saran untuk mahasiswa lain yang akan melakukan setup serupa

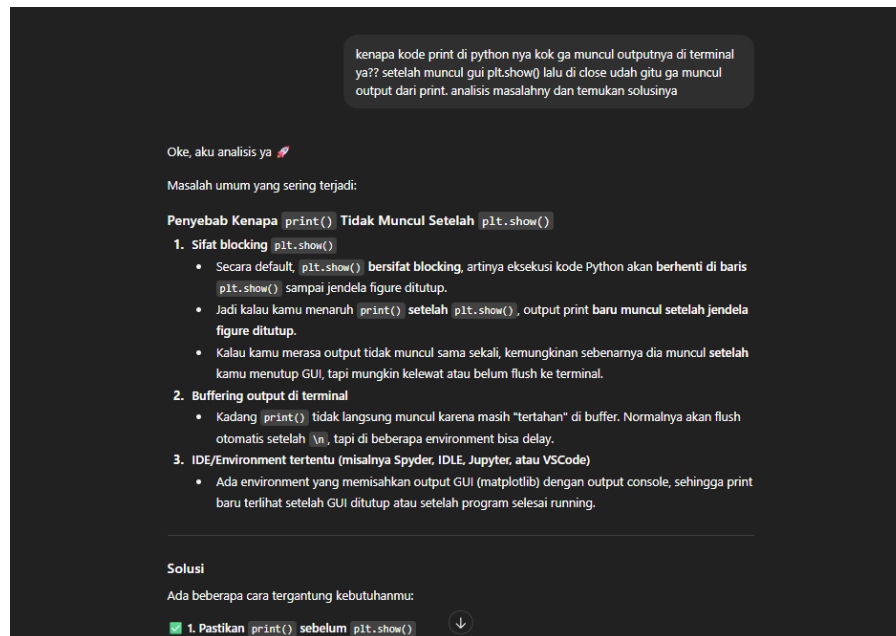
Jangan menyerah kalau ketemu baris merah (error) terus coba dan analisis masalahnya agar bisa belajar dan bisa menemukan solusinya. Selain itu, dokumentasikan setiap langkah yang diambil untuk memudahkan troubleshooting di masa depan.

Saya jadi paham fungsi dari environment management, cara membuat environment, menginstall library multimedia, dan melakukan verifikasi instalasi dengan test sederhana. Dengan kita pakai environment management kita bisa mengelola project dengan lebih baik dan terstruktur.

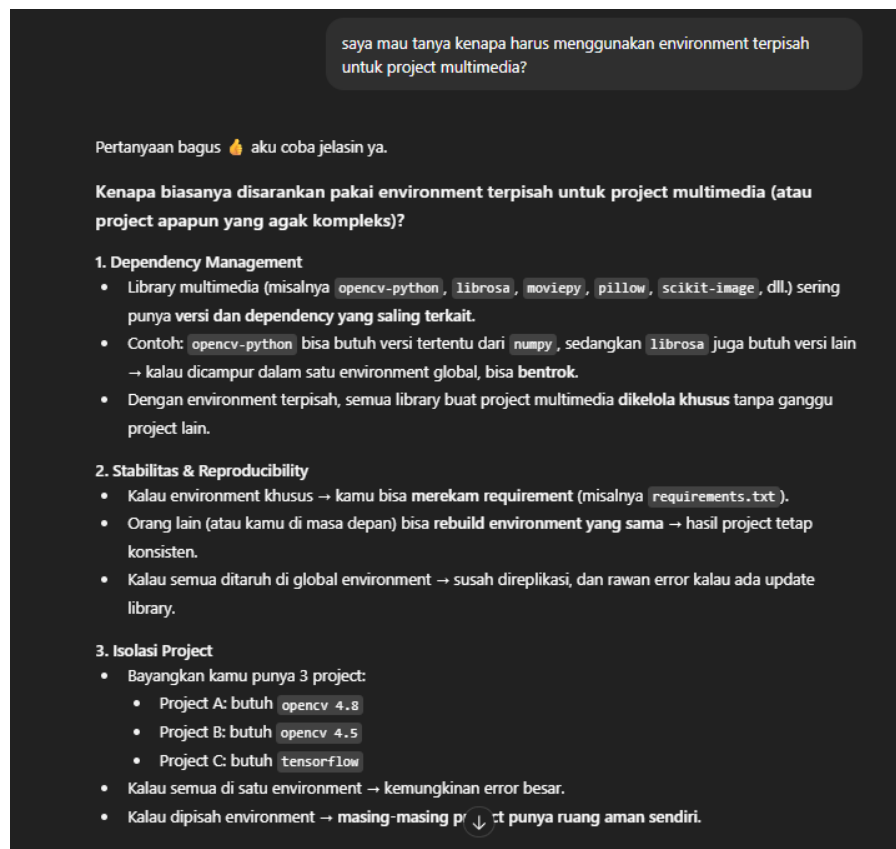
7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

- Link Chat GPT : [Multimedia Environment Setup](#)



Menganalisis masalah print output yang tidak keluar di terminal



Kenapa harus pakai envirointment terpisah?