



Nama: Isi Nama Di Sini (Dan NIM Di Sini)
Environment untuk Multimedia

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

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

Tanggal: November 18, 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: **Venv**
- Screenshot atau copy-paste output dari perintah verifikasi environment:

```
Transcript written on worksheet1.log.  
● > python -m venv .venv  
● > source .venv/bin/activate  
● > which python  
/home/fadilrifqi/Projects/mulmed/tugas/environment-setup/.venv/bin/python  
on master !10 ?4
```

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: **Venv dengan pip**
- Screenshot proses instalasi atau output sukses

```
> pip install opencv-python pillow scikit-image matplotlib
Collecting opencv-python
  Using cached opencv_python-4.12.0.88-cp37-abi3-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (11 kB)
Requirement already satisfied: pillow in ./venv/lib/python3.13/site-packages (11.3.0)
Collecting scikit-image
  Downloading scikit_image-0.25.2-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (11 kB)
Collecting matplotlib
  Using cached matplotlib-3.10.7-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (11 kB)
Collecting numpy<2.3.0,>=2 (from opencv-python)
  Using cached numpy-2.2.6-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (62 kB)
Requirement already satisfied: scipy>=1.11.4 in ./venv/lib/python3.13/site-packages (from scikit-image)
Collecting networkx>=3.0 (from scikit-image)
  Downloading networkx-3.5-py3-none-any.whl.metadata (6.3 kB)
Requirement already satisfied: imageio!=2.35.0,>=2.33 in ./venv/lib/python3.13/site-packages (from scikit-image)
  Using cached imageio-2.35.0-py3-none-any.whl.metadata (1.4 kB)
```

```
> pip install numpy pandas jupyter
Requirement already satisfied: numpy in ./venv/lib/python3.13/site-packages (2.3.5)
Collecting pandas
  Downloading pandas-2.3.3-cp313-cp313-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl.metadata (91 kB)
Collecting jupyter
  Downloading jupyter-1.1.1-py2.py3-none-any.whl.metadata (2.0 kB)
Collecting python-dateutil>=2.8.2 (from pandas)
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Using cached tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting notebook (from jupyter)
  Using cached notebook-7.4.7-py3-none-any.whl.metadata (10 kB)
Collecting jupyter-console (from jupyter)
```

```
> pip install librosa soundfile scipy
Collecting librosa
  Using cached librosa-0.11.0-py3-none-any.whl.metadata (8.7 kB)
Collecting soundfile
  Using cached soundfile-0.13.1-py2.py3-none-manylinux_2_28_x86_64.whl.metadata (16 kB)
Collecting scipy
  Using cached scipy-1.16.3-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (62 kB)
Collecting audioread>=2.1.9 (from librosa)
  Using cached audioread-3.1.0-py3-none-any.whl.metadata (9.0 kB)
Collecting numba>=0.51.0 (from librosa)
  Using cached numba-0.62.1-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (2.8 kB)
Collecting numpy>=1.22.3 (from librosa)
  Downloading numpy-2.3.5-cp313-cp313-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (62 kB)
Collecting scikit-learn>=1.1.0 (from librosa)
  Using cached scikit_learn-1.7.2-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (6.9 kB)
Collecting joblib>=1.0 (from librosa)
  Using cached joblib-1.5.2-py3-none-any.whl.metadata (5.6 kB)
```

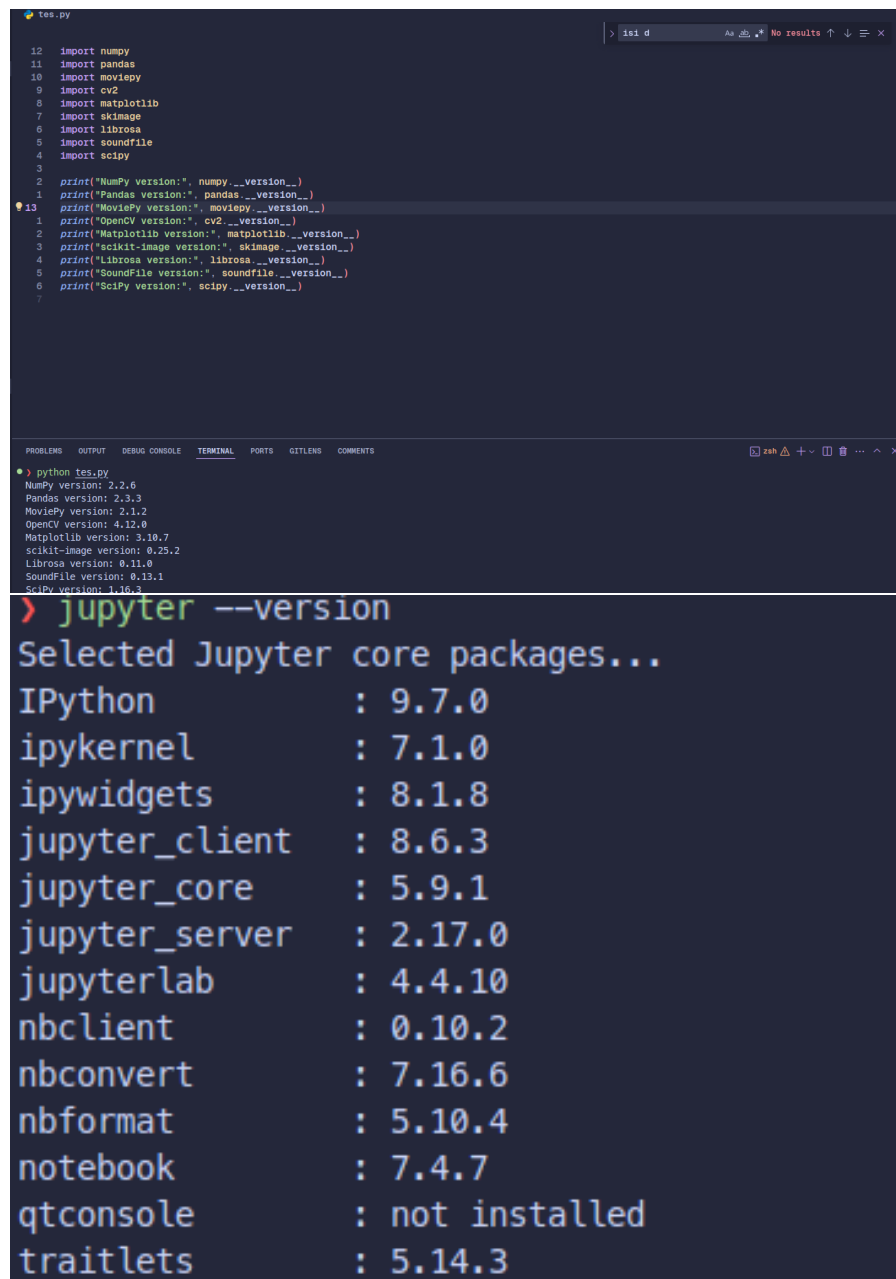
```
> pip install moviepy
Collecting moviepy
  Downloading moviepy-2.2.1-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: decorator<6.0,>=4.0.2 in ./venv/lib/python3.13/site-packages (from moviepy)
Collecting imageio<3.0,>=2.5 (from moviepy)
  Downloading imageio-2.37.2-py3-none-any.whl.metadata (9.7 kB)
Collecting imageio_ffmpeg>=0.2.0 (from moviepy)
  Downloading imageio_ffmpeg-0.6.0-py3-none-manylinux2014_x86_64.whl.metadata (1.5 kB)
Requirement already satisfied: numpy>=1.25.0 in ./venv/lib/python3.13/site-packages (from moviepy) (2.3.5)
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-1.2.1-py3-none-any.whl.metadata (25 kB)
Collecting pillow<12.0,>=9.2.0 (from moviepy)
  Downloading pillow-11.3.0-cp313-cp313-manylinux_2_27_x86_64.manylinux_2_28_x86_64.whl.metadata (9.0 kB)
```

- Daftar library yang berhasil diinstall dengan versinya
 - NumPy Versi 2.2.6
 - Pandas Versi 2.3.3
 - MoviePy Versi 2.1.2
 - OpenCV Versi 4.12.0
 - Matplotlib Versi 3.10.7
 - scikit-image Versi 0.25.2
 - Librosa Versi 0.11.0

- SoundFile Versi 0.13.1
- SciPy Versi 1.16.3
- Pillow Versi 11.3.0

3.4 Bagian 3: Verifikasi Instalasi

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



```

tes.py
12 import numpy
11 import pandas
10 import moviepy
9 import cv2
8 import matplotlib
7 import skimage
6 import librosa
5 import soundfile
4 import scipy
3
2 print("NumPy version:", numpy.__version__)
1 print("Pandas version:", pandas.__version__)
13 print("MoviePy version:", moviepy.__version__)
1 print("OpenCV version:", cv2.__version__)
2 print("Matplotlib version:", matplotlib.__version__)
3 print("scikit-image version:", skimage.__version__)
4 print("Librosa version:", librosa.__version__)
5 print("SoundFile version:", soundfile.__version__)
6 print("SciPy version:", scipy.__version__)
7

python tes.py
NumPy version: 2.2.6
Pandas version: 2.3.3
MoviePy version: 2.1.2
OpenCV version: 4.12.0
Matplotlib version: 3.10.7
scikit-image version: 0.25.2
Librosa version: 0.11.0
SoundFile version: 0.13.1
SciPy version: 1.16.3

jupyter --version
Selected Jupyter core packages...
IPython : 9.7.0
ipykernel : 7.1.0
ipywidgets : 8.1.8
jupyter_client : 8.6.3
jupyter_core : 5.9.1
jupyter_server : 2.17.0
jupyterlab : 4.4.10
nbclient : 0.10.2
nbconvert : 7.16.6
nbformat : 5.10.4
notebook : 7.4.7
qtconsole : not installed
traitlets : 5.14.3

```

Jalankan script dan dokumentasikan hasilnya:

3.5 Bagian 4: Simple Test dengan Sample Code

Buat dan jalankan contoh sederhana untuk setiap kategori multimedia:

3.5.1 Test Audio Processing

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

Kode 8: Test audio processing sederhana

3.5.2 Test Image Processing

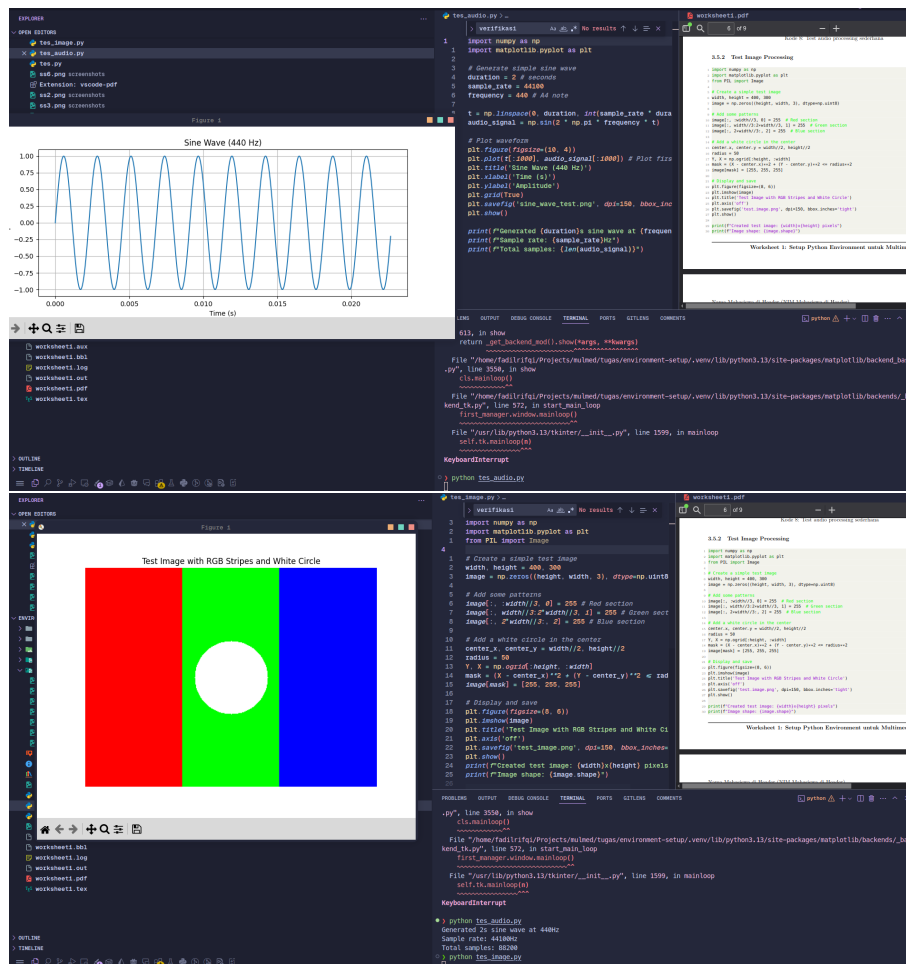
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from PIL import Image
4
5 # Create a simple test image
6 width, height = 400, 300
7 image = np.zeros((height, width, 3), dtype=np.uint8)
8
9 # Add some patterns
10 image[:, :width//3, 0] = 255 # Red section
11 image[:, width//3:2*width//3, 1] = 255 # Green section
12 image[:, 2*width//3:, 2] = 255 # Blue section
13
14 # Add a white circle in the center
15 center_x, center_y = width//2, height//2
16 radius = 50
17 Y, X = np.ogrid[:height, :width]
18 mask = (X - center_x)**2 + (Y - center_y)**2 <= radius**2
19 image[mask] = [255, 255, 255]
20
21 # Display and save
22 plt.figure(figsize=(8, 6))
23 plt.imshow(image)
24 plt.title('Test Image with RGB Stripes and White Circle')
25 plt.axis('off')
26 plt.savefig('test_image.png', dpi=150, bbox_inches='tight')
27 plt.show()
28
29 print(f"Created test image: {width}x{height} pixels")
30 print(f"Image shape: {image.shape}")
```

```
31 print(f"Image dtype: {image.dtype}")
```

Kode 9: Test image processing sederhana

Dokumentasikan hasil eksekusi:

- Screenshot output dari kedua script di atas
- Gambar yang dihasilkan (sine_wave_test.png dan test_image.png)



- Error message jika ada dan cara mengatasinya

4 Bagian Laporan

4.1 Output Verifikasi Instalasi

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

1 [PASTE OUTPUT DI SINI]

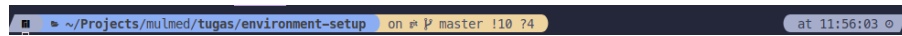
Kode 10: Output verifikasi instalasi

4.2 Screenshot Hasil Test

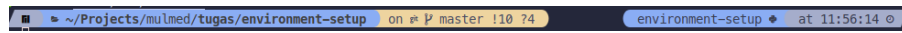
Sisipkan screenshot atau gambar hasil dari:

- Terminal/command prompt yang menunjukkan environment aktif

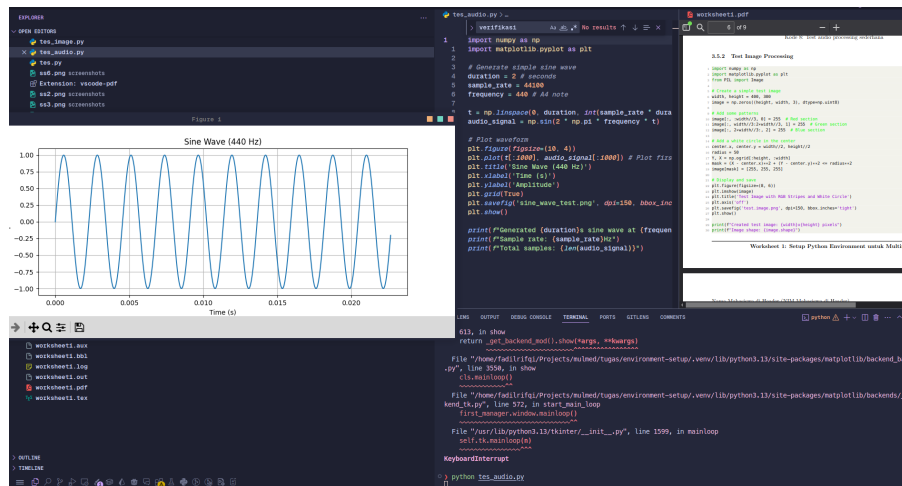
- sebelum aktivasi environment



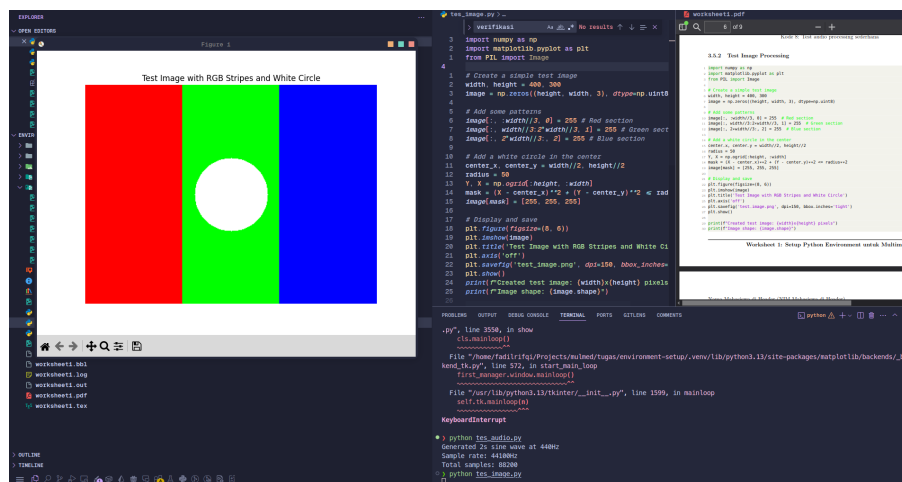
- setelah aktivasi environment



- Output dari script test audio (sine wave plot)



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



Gunakan perintah `|includegraphics` untuk menyisipkan gambar

4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

1. Mengapa penting menggunakan environment terpisah untuk project multimedia?
[Untuk mencegah konflik dependency antar library yang membutuhkan versi berbeda]
2. Apa perbedaan utama antara conda, venv, dan uv? Mengapa Anda memilih tool yang Anda gunakan?
[dari segi kecepatan dan library yang tersedia]

3. Library mana yang paling sulit diinstall dan mengapa?

[sejauh saya menjalankan instalasi di atas tidak ada error / kendala namun perbedaan nama library dan cara import menjadi kendala saat ingin tes seperti library pillow]

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

[mencari di dokumentasi resmi library tersebut]

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

- NumPy: Untuk mempermudah perhitungan seperti banyak fungsi matematika yang ada di numpy
- Pandas: Untuk manipulasi dan analisis data seperti csv xlsx dll
- MoviePy: Untuk manipulasi video
- OpenCV: Untuk pengolahan citra dan computer vision
- Matplotlib: Untuk visualisasi data
- scikit-image: Untuk pengolahan citra
- Librosa: Untuk analisis dan manipulasi audio
- SoundFile: Untuk membaca dan menulis file audio
- SciPy: Untuk komputasi ilmiah dan teknis
- Pillow: Untuk manipulasi gambar

4.4 Troubleshooting

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

- **Masalah 1:** *[Mendownload latex compiler]*

Solusi: *[Mencari tutorial di internet dan mengikuti langkah-langkahnya]*

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 numpy==2.2.6
2 pandas==2.3.3
3 moviepy==2.2.1
4 opencv-python==4.12.0.88
5 matplotlib==3.10.7
6 scikit-image==0.25.2
7 librosa==0.11.0
8 soundfile==0.13.1
9 scipy==1.16.3
10 pillow==11.3.0
11 jupyter==1.1.1
```

Kode 13: Environment/Requirements file

6 Kesimpulan

Tuliskan kesimpulan Anda mengenai:

- Pengalaman setup Python environment untuk multimedia : Cukup mudah diikuti langkah-langkahnya
- Persiapan untuk project multimedia selanjutnya : Dengan adanya environment yang sudah terpisah ini diharapkan tidak ada konflik dependency di project selanjutnya
- Saran untuk mahasiswa lain yang akan melakukan setup serupa : Jika menemukan kesulitan instalasi disarankan untuk membaca dokumentasi resmi terlebih dahulu karena biasanya sudah lengkap caranya disitu

[Kesimpulan Anda di sini]

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

- <https://tug.org/texlive/quickinstall.html>
- <https://pillow.readthedocs.io/en/stable/handbook/tutorial.html>

References