



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: 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

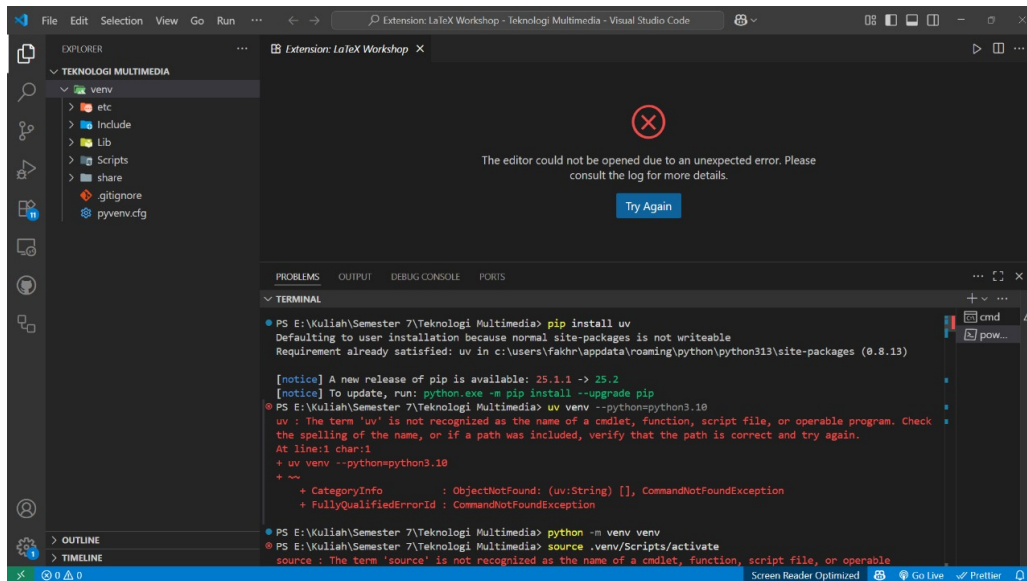
3.2.3 Opsi 3: Menggunakan uv (Modern dan cepat)

```
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



Gambar 1: Install UV dan Aktivasi Environment

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

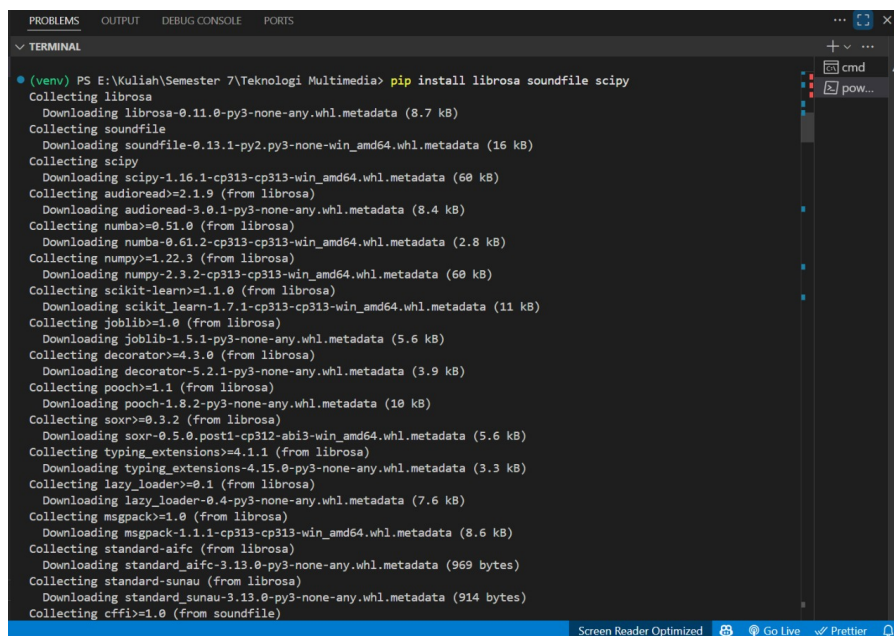
3.3.4 Library General Purpose

```
1 # Untuk conda:  
2 conda install numpy pandas jupyter  
3  
4 # Untuk pip (venv/uv):  
5 pip install numpy pandas jupyter
```

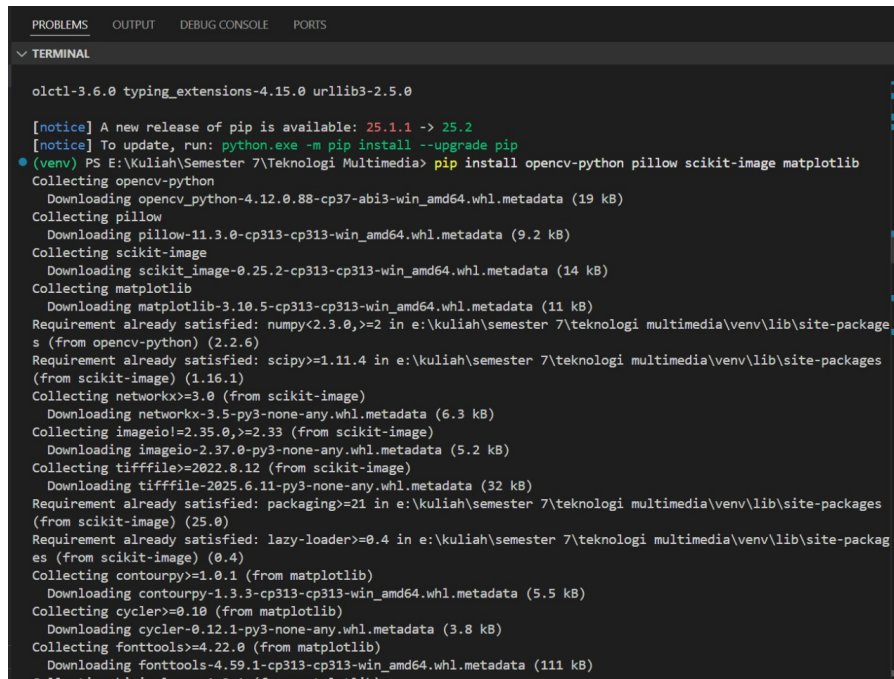
Kode 7: Instalasi library umum

Dokumentasikan di sini:

- Perintah instalasi yang Anda gunakan
- Screenshot proses instalasi atau output sukses
- Daftar library yang berhasil diinstall dengan versinya



Gambar 1. Install Audio Library



```

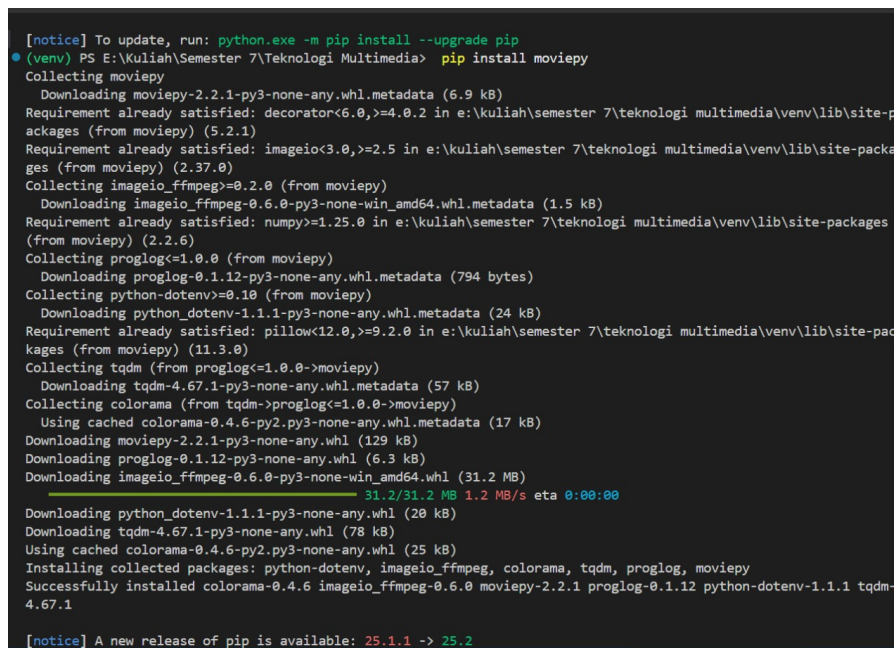
PROBLEMS OUTPUT DEBUG CONSOLE PORTS
TERMINAL

olc1t1-3.6.0 typing_extensions-4.15.0 urllib3-2.5.0

[notice] A new release of pip is available: 25.1.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS E:\Kuliah\Semester 7\Teknologi Multimedia> pip install opencv-python pillow scikit-image matplotlib
Collecting opencv-python
  Downloading opencv_python-4.12.0.88-cp37-abi3-win_amd64.whl.metadata (19 kB)
Collecting pillow
  Downloading pillow-11.3.0-cp313-cp313-win_amd64.whl.metadata (9.2 kB)
Collecting scikit-image
  Downloading scikit_image-0.25.2-cp313-cp313-win_amd64.whl.metadata (14 kB)
Collecting matplotlib
  Downloading matplotlib-3.10.5-cp313-cp313-win_amd64.whl.metadata (11 kB)
Requirement already satisfied: numpy<2.3.0,>=2 in e:\kuliah\semester 7\teknologi multimedia\venv\lib\site-packages (from opencv-python) (2.2.6)
Requirement already satisfied: scipy>=1.11.4 in e:\kuliah\semester 7\teknologi multimedia\venv\lib\site-packages (from scikit-image) (1.16.1)
Collecting networkx>=3.0 (from scikit-image)
  Downloading networkx-3.5-py3-none-any.whl.metadata (6.3 kB)
Collecting imageio<2.35.0,>=2.33 (from scikit-image)
  Downloading imageio-2.37.0-py3-none-any.whl.metadata (5.2 kB)
Collecting tifffile>=2022.8.12 (from scikit-image)
  Downloading tifffile-2025.6.11-py3-none-any.whl.metadata (32 kB)
Requirement already satisfied: packaging>=21 in e:\kuliah\semester 7\teknologi multimedia\venv\lib\site-packages (from scikit-image) (25.0)
Requirement already satisfied: lazy-loader>=0.4 in e:\kuliah\semester 7\teknologi multimedia\venv\lib\site-packages (from scikit-image) (0.4)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.3-cp313-cp313-win_amd64.whl.metadata (5.5 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.59.1-cp313-cp313-win_amd64.whl.metadata (111 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)

```

Gambar 2. Install Image Library



```

[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS E:\Kuliah\Semester 7\Teknologi Multimedia> 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 e:\kuliah\semester 7\teknologi multimedia\venv\lib\site-packages (from moviepy) (5.2.1)
Requirement already satisfied: imageio<3.0,>=2.5 in e:\kuliah\semester 7\teknologi multimedia\venv\lib\site-packages (from moviepy) (2.37.0)
Collecting imageio_ffmpeg>=0.2.0 (from moviepy)
  Downloading imageio_ffmpeg-0.6.0-py3-none-win_amd64.whl.metadata (1.5 kB)
Requirement already satisfied: numpy>=1.25.0 in e:\kuliah\semester 7\teknologi multimedia\venv\lib\site-packages (from moviepy) (2.2.6)
Collecting proglog<=1.0.0 (from moviepy)
  Downloading proglog-0.1.12-py3-none-any.whl.metadata (794 bytes)
Collecting python-dotenv>=0.10 (from moviepy)
  Downloading python_dotenv-1.1.1-py3-none-any.whl.metadata (24 kB)
Requirement already satisfied: pillow<12.0,>=9.2.0 in e:\kuliah\semester 7\teknologi multimedia\venv\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)
Collecting colorama (from tqdm->proglog<=1.0.0->moviepy)
  Using cached colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Downloaded moviepy-2.2.1-py3-none-any.whl (129 kB)
Downloaded proglog-0.1.12-py3-none-any.whl (6.3 kB)
Downloaded imageio_ffmpeg-0.6.0-py3-none-win_amd64.whl (31.2 MB)
31.2/31.2 MB 1.2 MB/s eta 0:00:00
Downloaded python_dotenv-1.1.1-py3-none-any.whl (20 kB)
Downloaded tqdm-4.67.1-py3-none-any.whl (78 kB)
Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: python-dotenv, imageio_ffmpeg, colorama, tqdm, proglog, moviepy
Successfully installed 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

[notice] A new release of pip is available: 25.1.1 -> 25.2

```

Gambar 3. Install Video Library

```

[notice] A new release of pip is available: 25.1.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
(venv) PS E:\Kuliah\Semester 7\Teknologi Multimedia> pip install numpy pandas jupyter
Requirement already satisfied: numpy in e:\kuliah\semester 7\teknologi multimedia\venv\lib\site-packages (2.2.6)
Collecting pandas
  Downloading pandas-2.3.2-cp313-cp313-win_amd64.whl.metadata (19 kB)
Collecting jupyter
  Downloading jupyter-1.1.1-py2.py3-none-any.whl.metadata (2.0 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in e:\kuliah\semester 7\teknologi multimedia\venv\lib\site-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting notebook (from jupyter)
  Downloading notebook-7.4.5-py3-none-any.whl.metadata (10 kB)
Collecting jupyter-console (from jupyter)
  Downloading jupyter_console-6.6.3-py3-none-any.whl.metadata (5.8 kB)
Collecting nbconvert (from jupyter)
  Downloading nbconvert-7.16.6-py3-none-any.whl.metadata (8.5 kB)
Collecting ipykernel (from jupyter)
  Downloading ipykernel-6.30.1-py3-none-any.whl.metadata (6.2 kB)
Collecting ipywidgets (from jupyter)
  Downloading ipywidgets-8.1.7-py3-none-any.whl.metadata (2.4 kB)
Collecting jupyterlab (from jupyter)
  Downloading jupyterlab-4.4.6-py3-none-any.whl.metadata (16 kB)
Requirement already satisfied: six>=1.5 in e:\kuliah\semester 7\teknologi multimedia\venv\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Collecting comm>=0.1.1 (from ipykernel->jupyter)
  Downloading comm-0.2.3-py3-none-any.whl.metadata (3.7 kB)
Collecting debugpy>=1.6.5 (from ipykernel->jupyter)
  Downloading debugpy-1.8.16-cp313-cp313-win_amd64.whl.metadata (1.4 kB)
Collecting ipython>=7.23.1 (from ipykernel->jupyter)

```

Gambar 4. Install General Library

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

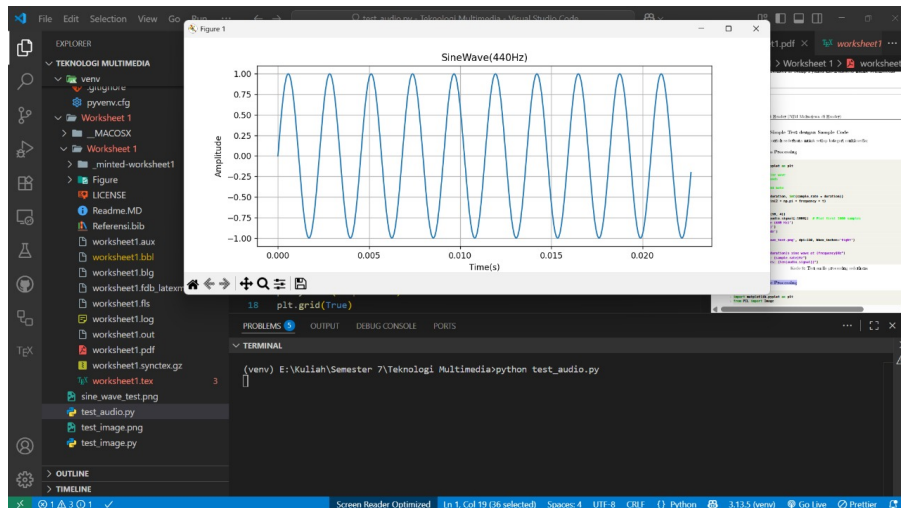
3.5.2 Test Image Processing

```
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}")
```

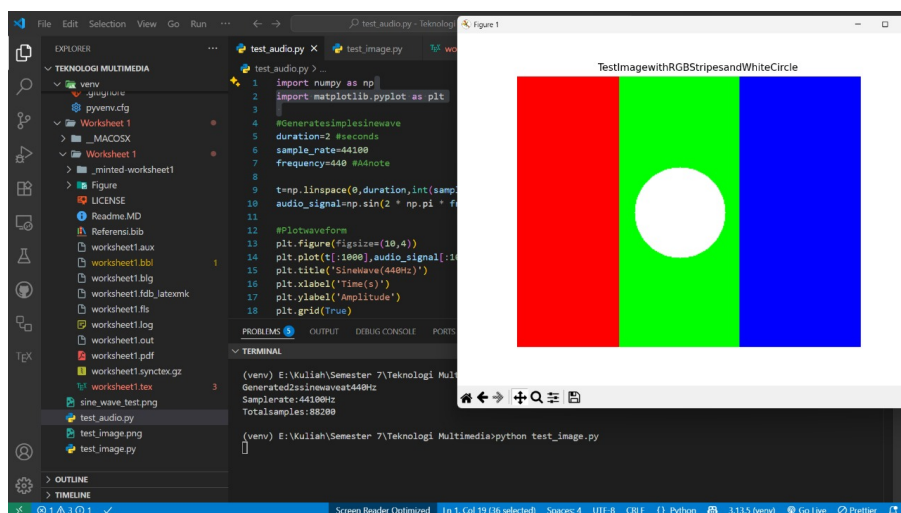
Kode 9: Test image processing sederhana

Dokumentasikan hasil eksekusi:

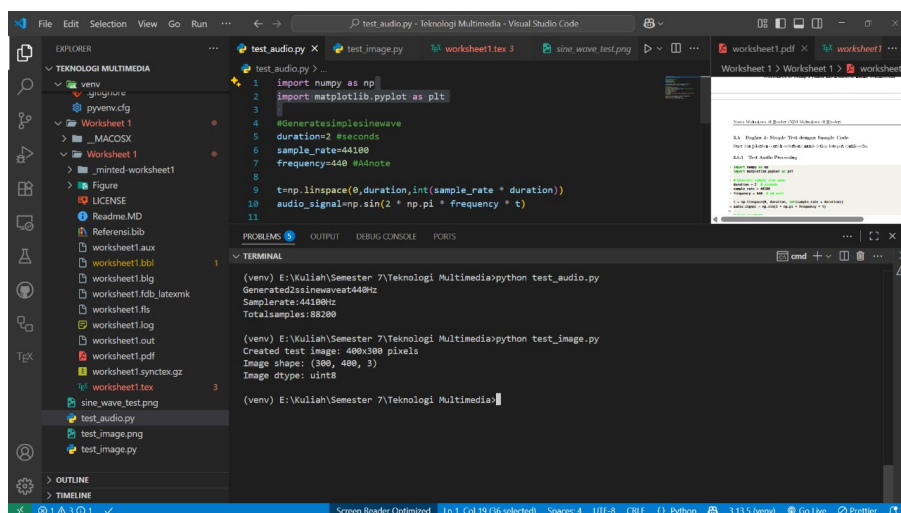
- Screenshot output dari kedua script di atas
- Gambar yang dihasilkan (sine_wave_test.png dan test_image.png)
- Error message jika ada dan cara mengatasinya



Gambar 5. Test Audio Output



Gambar 6. Test image Output



Gambar 6. Output dari script test image dan audio

4 Bagian Laporan

4.1 Output Verifikasi Instalasi

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

```

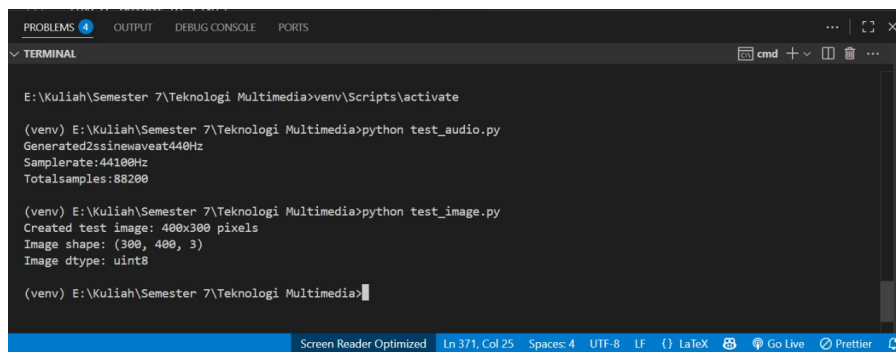
1 [(venv) E:\Kuliah\Semester 7\Teknologi Multimedia>python test_audio.py
2 Generated2ssinewaveat440Hz
3 Samplerate:44100Hz
4 Totalsamples:88200
5 Generated2ssinewaveat440Hz
6 Samplerate:44100Hz
7 Totalsamples:88200
8
9 (venv) E:\Kuliah\Semester 7\Teknologi Multimedia>python test_image.py
10 Created test image: 400x300 pixels
11 Image shape: (300, 400, 3)
12 Image dtype: uint8
13 Generated2ssinewaveat440Hz
14 Samplerate:44100Hz
15 Totalsamples:88200]
```

Kode 10: Output verifikasi instalasi

4.2 Screenshot Hasil Test

Sisipkan screenshot atau gambar hasil dari:

- Terminal/command prompt yang menunjukkan environment aktif
- Output dari script test audio (sine wave plot)
- Output dari script test image (RGB stripes dengan circle)



Gambar 6. environment aktif dan output dari script test audio dan image

Gunakan perintah `\includegraphics` untuk menyisipkan gambar

4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

1. Mengapa penting menggunakan environment terpisah untuk project multimedia?

karena setiap project biasanya punya kebutuhan library dan versinya masing-masing. Kalau semua project dijalankan di satu environment yang sama, bisa terjadi versi nya tidak sama

2. Perbedaan nya ini saya tangkap dari bapak waktu di kelas, Conda biasanya lebih lama saat proses instalasi, tetapi kelebihanannya adalah library yang sudah pernah diinstal bisa dipakai ulang di project lain tanpa perlu mengunduh lagi. Kalo UV prosesnya sangat cepat karena dioptimalkan untuk manajemen dependency modern, tetapi setiap project harus menginstal library dari awal sehingga agak boros waktu dan ruang. Venv sendiri hanya membuat environment virtual tanpa mengatur paket bawaan, jadi setiap kali membuat environment baru kita perlu instal library lagi melalui pip.

Jawaban Anda di sini

3. Library mana yang paling sulit diinstall dan mengapa?

Untuk saya sendiri selama proses instalasi library tidak ada kesulitan dan tidak error

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

cara mengatasi dependency conflict biasanya dengan membuat environment baru dan menginstal library satu per satu, atau menggunakan conda yang bisa mengelola dependency dengan lebih baik

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

Librosa untuk audio processing, soundfile untuk membaca dan menulis file audio, scipy untuk komputasi ilmiah, opencv untuk image processing, pillow untuk manipulasi gambar, scikit-image untuk analisis gambar, matplotlib untuk visualisasi data, ffmpeg untuk video processing, moviepy untuk editing video, numpy untuk komputasi numerik, pandas untuk manipulasi data, jupyter untuk notebook interaktif

4.4 Troubleshooting

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

- **Masalah 1:** *Tidak Bisa Menginstall UV dari Web Pak martin*

Solusi: Saya tanya ke chat gpt, dan chat gpt nya mengasih kode untuk install uv dan cara mengaktifkan environment uv

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 audiop-lts==0.2.2
9 audioread==3.0.1
10 babel==2.17.0
11 beautifulsoup4==4.13.5
12 bleach==6.2.0
13 certifi==2025.8.3
14 cffi==1.17.1
15 charset-normalizer==3.4.3
16 colorama==0.4.6
17 comm==0.2.3
18 contourpy==1.3.3
19 cycler==0.12.1
20 debugpy==1.8.16
21 decorator==5.2.1
22 defusedxml==0.7.1
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==9.4.0
35 ipython_pygments_lexers==1.1.1
36 ipywidgets==8.1.7
37 isoduration==20.11.0
38 jedi==0.19.2
39 Jinja2==3.1.6
40 joblib==1.5.1
41 json5==0.12.1
42 jsonpointer==3.0.0
43 jsonschema==4.25.1
44 jsonschema-specifications==2025.4.1
45 jupyter==1.1.1
46 jupyter-console==6.6.3
47 jupyter-events==0.12.0
48 jupyter-lsp==2.2.6
49 jupyter_client==8.6.3
50 jupyter_core==5.8.1
51 jupyter_server==2.17.0
52 jupyter_server_terminals==0.5.3
53 jupyterlab==4.4.6
54 jupyterlab_pygments==0.3.0
55 jupyterlab_server==2.27.3
56 jupyterlab_widgets==3.0.15
57 kiwisolver==1.4.9
58 lark==1.2.2
59 lazy_loader==0.4
60 librosa==0.11.0
61 llvmlite==0.44.0
62 MarkupSafe==3.0.2
```

```
63 matplotlib==3.10.5
64 matplotlib-inline==0.1.7
65 mistune==3.1.3
66 moviepy==2.2.1
67 msgpack==1.1.1
68 nbclient==0.10.2
69 nbconvert==7.16.6
70 nbformat==5.10.4
71 nest-asyncio==1.6.0
72 networkx==3.5
73 notebook==7.4.5
74 notebook_shim==0.2.4
75 numba==0.61.2
76 numpy==2.2.6
77 opencv-python==4.12.0.88
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==1.16.1
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 standard-aifc==3.13.0
119 standard-chunk==3.13.0
120 standard-sunau==3.13.0
121 terminado==0.18.1
122 threadpoolctl==3.6.0
123 tifffile==2025.6.11
124 tinycss2==1.4.0
```

```
125 tornado==6.5.2
126 tqdm==4.67.1
127 traitlets==5.14.3
128 types-python-dateutil==2.9.0.20250822
129 typing_extensions==4.15.0
130 tzdata==2025.2
131 uri-template==1.3.0
132 urllib3==2.5.0
133 wcwidth==0.2.13
134 webcolors==24.11.1
135 webencodings==0.5.1
136 websocket-client==1.8.0
137 widgetsnbextension==4.0.14
```

Kode 13: 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

Dari pengalaman setup Python environment untuk multimedia, saya jadi lebih paham mengenai apa itu framework serta berbagai library yang digunakan untuk audio processing, image processing, dan lain-lain. Proses ini membuat saya memiliki dasar yang cukup kuat, sehingga untuk project multimedia berikutnya saya sudah siap dan hanya perlu menunggu tahapan implementasinya. Untuk mahasiswa lain yang akan melakukan setup serupa, perlu diperhatikan bahwa ada kemungkinan beberapa laptop tidak bisa menginstal uv atau mengalami kendala saat aktivasi dari web yang dibagikan oleh Pak Martin. Jika menghadapi masalah seperti itu, solusi yang bisa dilakukan adalah mencari referensi tambahan, termasuk memanfaatkan bantuan ChatGPT untuk menemukan jalan keluarnya.

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

[Link Jawaban ChatGPT](#)

References