



Nama: **Bintang Fikri Fauzan** 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:

```
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

```
● (mm1) PS D:\Code\Sem 7\Mulmed> uv venv mm1
Using CPython 3.10.16
Creating virtual environment at: mm1
Activate with: mm1\Scripts\activate
○ (mm1) PS D:\Code\Sem 7\Mulmed>
```

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:

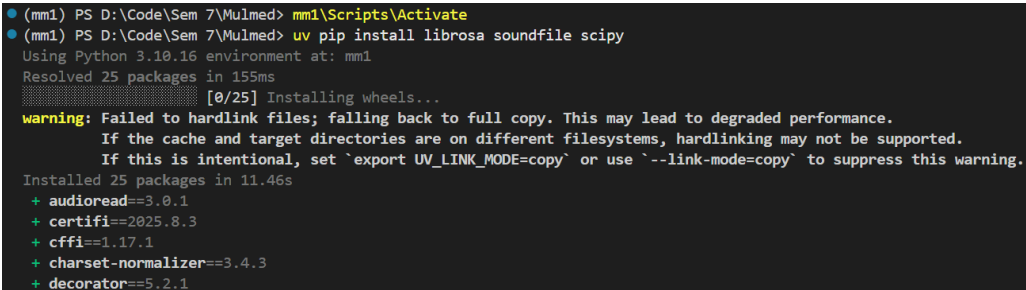
- Perintah instalasi yang Anda gunakan

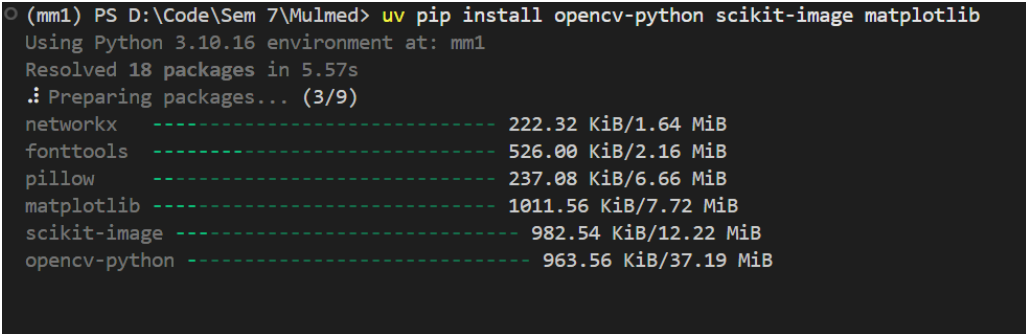
```

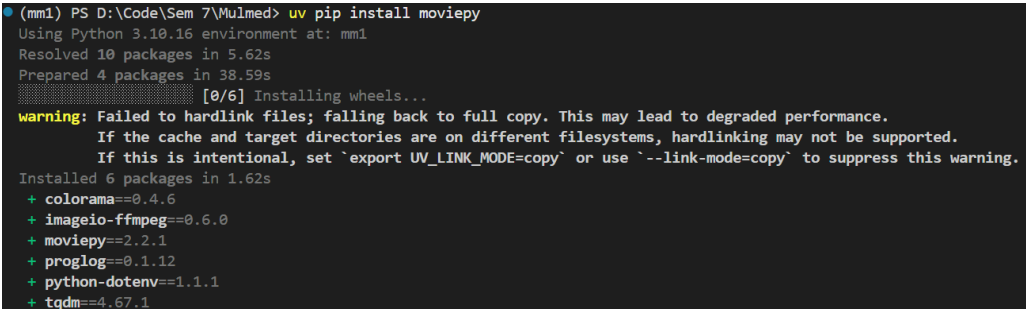
1 uv pip install librosa soundfile scipy
2 uv pip install opencv-python pillow scikit-image matplotlib
3 uv pip install moviepy
4 uv pip install numpy pandas jupyter
5

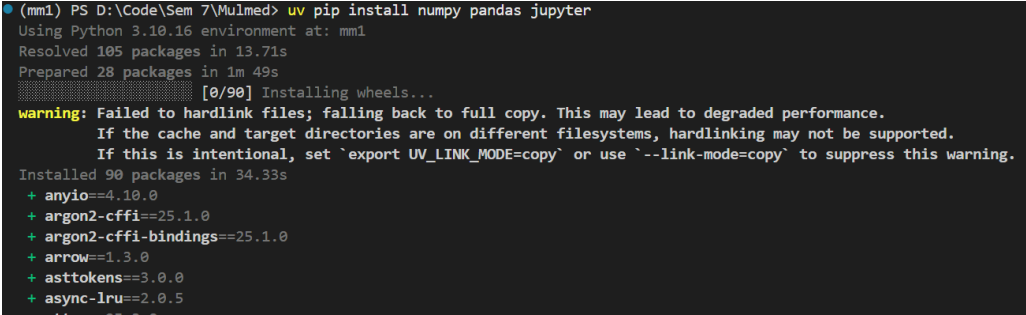
```

- Screenshot proses instalasi atau output sukses

1. 

2. 

3. 

4. 

- Daftar library yang berhasil diinstall dengan versinya

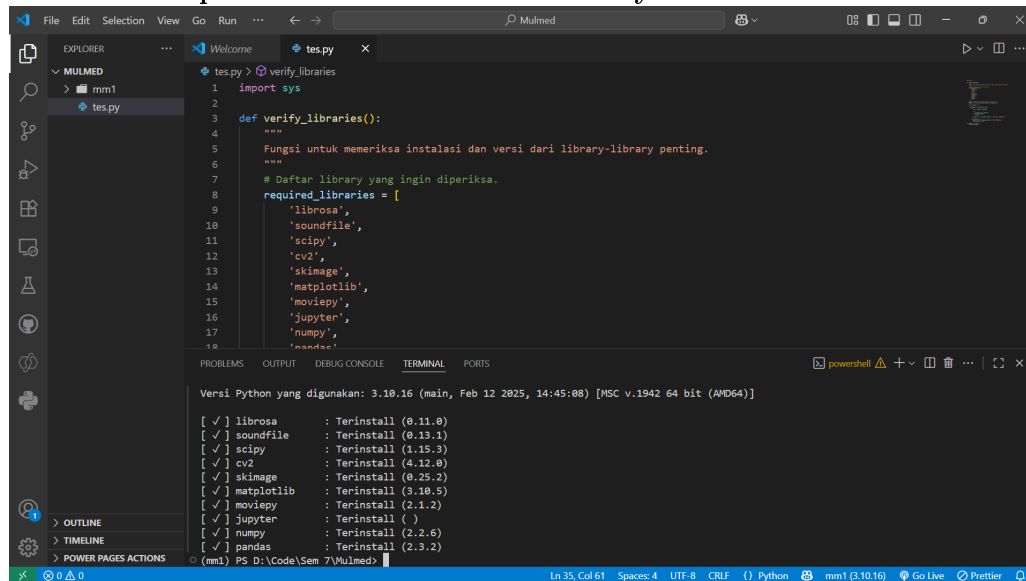
1. librosa: 0.11.0
2. soundfile: 0.13.1
3. scipy: 1.15.3
4. opencv-python: 4.12.0.88
5. scikit-image: 0.25.2

6. matplotlib: 3.10.5
7. moviepy: 2.2.1
8. numpy: 2.2.6
9. pandas: 2.3.2
10. jupyter: 1.1.1

3.4 Bagian 3: Verifikasi Instalasi

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

Jalankan script dan dokumentasikan hasilnya:



```

tes.py > verify_libraries
1 import sys
2
3 def verify_libraries():
4     """
5     Fungsi untuk memeriksa instalasi dan versi dari library-library penting.
6     """
7     # Daftar library yang ingin diperiksa.
8     required_libraries = [
9         'librosa',
10        'soundfile',
11        'scipy',
12        'cv2',
13        'skimage',
14        'matplotlib',
15        'moviepy',
16        'jupyter',
17        'numpy',
18        'pandas'
19    ]
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

Versi Python yang digunakan: 3.10.16 (main, Feb 12 2025, 14:45:08) [MSC v.1942 64 bit (AMD64)]

[✓] librosa : Terinstall (0.11.0)
[✓] soundfile : Terinstall (0.13.1)
[✓] scipy : Terinstall (1.15.3)
[✓] cv2 : Terinstall (4.12.0)
[✓] skimage : Terinstall (0.25.2)
[✓] matplotlib : Terinstall (3.10.5)
[✓] moviepy : Terinstall (2.1.2)
[✓] jupyter : Terinstall ( )
[✓] numpy : Terinstall (2.2.6)
[✓] pandas : Terinstall (2.3.2)

```

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 Audio Signal Info:
2 Generated 2s sine wave at 440Hz
3 Sample rate: 44100Hz
4 Total samples: 88200
5

```

```
6 Image Info:
7 Created test image: 400x300 pixels
8 Image shape: (300, 400, 3)
9 Image dtype: uint8
```

Kode 10: Output verifikasi instalasi

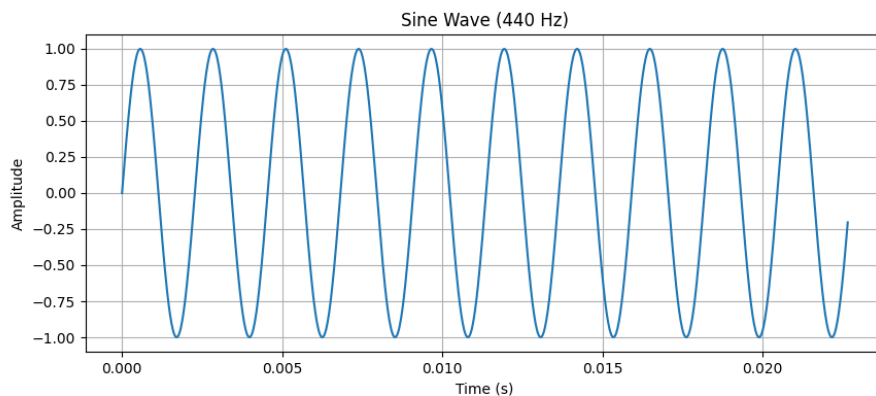
4.2 Screenshot Hasil Test

Sisipkan screenshot atau gambar hasil dari:

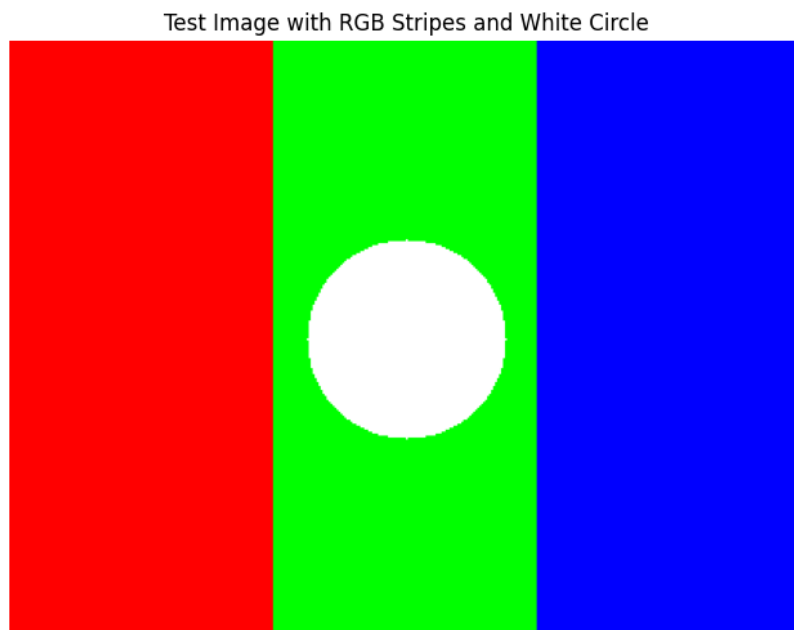
- Terminal/command prompt yang menunjukkan environment aktif

```
● PS D:\Code\Sem 7\Mulmed> mm1\Scripts\Activate
❖❖ (mm1) PS D:\Code\Sem 7\Mulmed> █
```

- Output dari script test audio (sine wave plot)



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



4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

1. Mengapa penting menggunakan environment terpisah untuk project multimedia?

Menggunakan environment terpisah penting untuk menghindari konflik antara dependensi yang mungkin dibutuhkan oleh proyek multimedia yang berbeda. Dengan memisahkan lingkungan, kita dapat memastikan bahwa setiap proyek memiliki paket dan versi yang tepat tanpa saling mempengaruhi.

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

- *Conda: ukuran environment yang besar dan banyak library prebuild.*
- *venv: built-in Python, ringan, tapi perlu install library manual.*
- *uv: lightweight, fokus pada kecepatan, tapi lebih sedikit fitur.*

Saya memilih uv karena ringan dan cepat, serta cukup untuk kebutuhan project multimedia saya.

3. Library mana yang paling sulit diinstall dan mengapa?

Saya tidak merasa ada library tertentu yang sulit diinstall.

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

Update library ke versi terbaru atau downgrade ke versi yang lebih stabil.

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

- *soundfile: digunakan untuk membaca dan menulis file audio dalam berbagai format.*
- *librosa: digunakan untuk menganalisis audio terutama berkaitan dengan musik.*
- *numpy: digunakan untuk pengelolaan komputasi numerik dasar dan efisien.*
- *pandas: digunakan untuk memanipulasi serta menganalisis data secara terstruktur.*
- *scipy: digunakan untuk pengelolaan fungsi algoritma matematika yang lebih spesifik dan kompleks.*
- *matplotlib: digunakan untuk visualisasi data dalam bentuk grafik dan plot.*
- *moviepy: digunakan untuk mengedit dan memproses file video.*
- *opencv: digunakan untuk pengolahan citra dan visi komputer.*
- *scikit-image: digunakan untuk algoritma pengolahan citra.*
- *jupyter: digunakan untuk membuat dan berbagi dokumen yang berisi kode, persamaan, visualisasi, dan teks naratif.*

4.4 Troubleshooting

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

- **Tidak Ada Masalah**

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==4.3.8
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

Pengalaman setup Python environment untuk multimedia memberikan saya pemahaman yang baik tentang bagaimana lingkungan pengembangan multimedia berbasis Python dapat dijalankan. Dengan adanya environment yang terisolasi, saya dapat mengelola dependensi dan paket dengan mudah tanpa mengganggu sistem yang ada. Untuk project multimedia selanjutnya dapat disiapkan berupa dependensi yang lebih spesifik sesuai dengan kebutuhan project yang ditugaskan kedepannya.

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

[Link Percakapan dengan LLM](#)

References