



Nama: **Lucky Immanuel Sitanggang (122140179)** 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 L<sup>A</sup>T<sub>E</sub>X

## 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 L<sup>A</sup>T<sub>E</sub>X 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

|   |
|---|
| (multimedia-uv) PS D:\Se\Multimedia> where python                                   |
| (multimedia-uv) PS D:\Se\Multimedia> Get-Command python                             |
| CommandType Name Version Source   |
| ----- ----- ----- -----   |
| Application python.exe 3.10.18... D:\Se\Multimedia\multimedia-uv\Scripts\python.exe |

Gambar 1: Output verifikasi environment Python

### 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

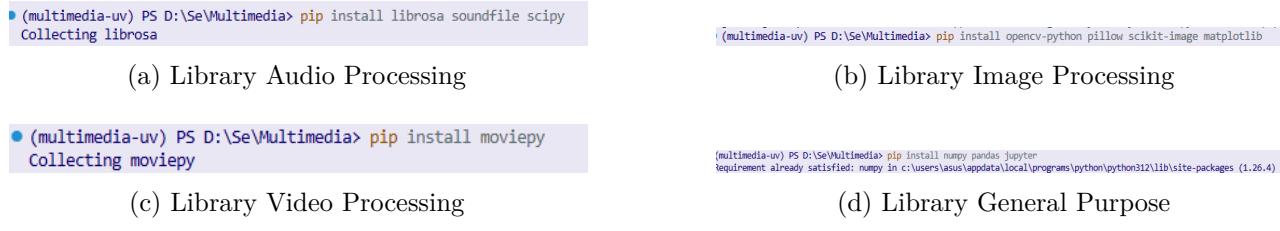
#### 3.3.4 Library General Purpose

```
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



Gambar 2: Installation Commands

- Screenshot proses instalasi atau output sukses

#### (a) Process Install Library Audio Processing

(b) Process Install Library Image Processing

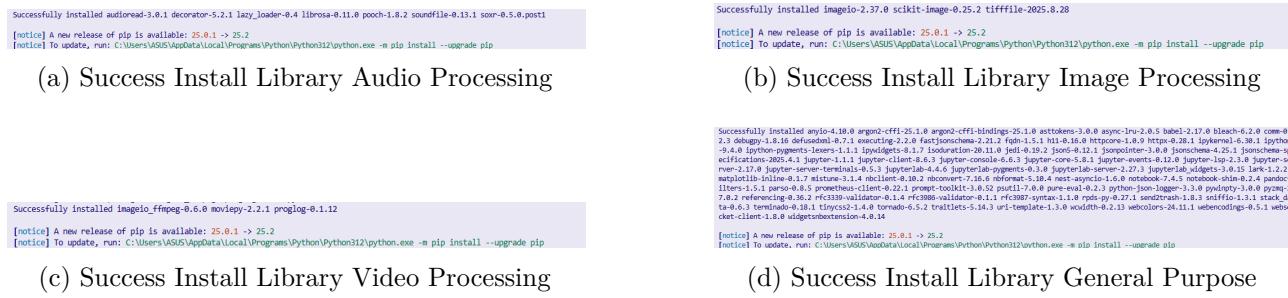
```
Collecting soupsieve
  Downloading soupsieve-2.2.1-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: decorator>=4.0.2 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from moviepy) (5.4.0)
Requirement already satisfied: imgkit<0.8,>=0.7.5 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from moviepy) (0.8.0)
Requirement already satisfied: pillow>2.0,>=2.0.0 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from moviepy) (2.3.7)
Collecting Imageio
  Downloading imageio_ffmpeg-0.6.0-py3-none-anywhl.metadata (1.5 kB)
Requirement already satisfied: imageio>=0.6.0 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from moviepy) (0.6.0)
Collecting prolog>=1.0.0 (from moviepy)
  Downloading prolog-0.1.12-py3-none-any.whl.metadata (794 bytes)
Requirement already satisfied: python>=3.6 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from moviepy) (1.1.4)
Requirement already satisfied: pilLOW>2.0,>=2.0.0 in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from moviepy) (10.4.1)
Requirement already satisfied: type>c in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from prolog<1.0.0,>moviepy) (4.67.0)
Requirement already satisfied: colorama in c:\users\asus\appdata\local\programs\python\python312\lib\site-packages (from type>c>prolog<1.0.0->moviepy) (4.1.3)
Downloaded: module>2.2.1-py3-none-any.whl (120 kB)
Downloaded: imgkit>0.6.0-py3-none-anywhl (31.2 MB)
Downloaded: Imageio>imageio_ffmpeg-0.6.0-py3-none-anywhl (6.1 kB)
  Downloading prolog-0.1.12-py3-none-any.whl (6.1 kB)
  Downloaded: prolog-0.1.12-py3-none-any.whl (6.1 kB)
  100%|██████████| 6.1 kB 125.4 kB / eta 00:00:00
Installing collected packages: imageio_ffmpeg, prolog, moviepy
```

### (c) Process Install Library Video Processing

(d) Process Install Library General Purpose

Gambar 3: Installation Process

- Daftar library yang berhasil diinstall dengan versinya



Gambar 4: Successful Installation

### 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

## 4 Bagian Laporan

### 4.1 Output Verifikasi Instalasi

Copy-paste output lengkap dari script **test\_multimedia.py** di sini:

```

1 (multimedia-uv) PS D:\Se\Multimedia> python -u "d:\Se\Multimedia\Tugas1\test_multimedia.py"
2 STARTING COMPREHENSIVE LIBRARY TEST
3 Test Date: 2025-08-29 18:43:25
4 Python Version: 3.10.18 (main, Aug 18 2025, 19:15:11) [MSC v.1944 64 bit (AMD64)]
5 =====
6 1. TESTING GENERAL PURPOSE LIBRARIES
7 =====
8
9 Testing NumPy...
10 NumPy array shape: (5, 3)
11 Array mean: 0.5686
12 Array max: 0.8726
13 Matrix multiplication result shape: (5, 5)
14
15 Testing Pandas...
16 DataFrame created with shape: (5, 4)
17 Average age: 30.0
18 Average salary: Rp 6,400,000
19 Group by operations completed
20 General Purpose libraries test PASSED

```

```
21=====
23 2. TESTING AUDIO PROCESSING LIBRARIES
24 =====
25 All audio libraries imported successfully
26
27 Testing Audio Processing...
28 Generated audio signal: 44100 samples at 22050 Hz
29 Spectral centroid calculated: mean = 566.90 Hz
30 MFCC features extracted: shape (13, 87)
31 Chroma features extracted: shape (12, 87)
32 Applied low-pass filter (cutoff: 1000 Hz)
33 FFT analysis: dominant frequency = 440.0 Hz
34 SoundFile library available for audio I/O operations
35 Audio Processing libraries test PASSED
36
37 =====
38 3. TESTING IMAGE PROCESSING LIBRARIES
39 =====
40 All image processing libraries imported successfully
41
42 Testing Image Processing...
43 Created synthetic image: (200, 200, 3)
44 OpenCV: Converted to grayscale and detected edges
45 OpenCV: Applied Gaussian blur
46 PIL: Resized to (100, 100) and rotated 45
47 Scikit-image: Canny edge detection
48 Scikit-image: Gaussian filter applied
49 Scikit-image: Found 7 regions in binary image
50 Matplotlib: Created image processing visualization
51 Image Processing libraries test PASSED
52
53 =====
54 4. TESTING VIDEO PROCESSING LIBRARIES
55 =====
56 MoviePy imported successfully
57
58 Testing Video Processing...
59 Created synthetic video clips (2 seconds each at 24 fps)
60 - FPS: 24
61 Video Processing libraries test PASSED
62
63 =====
64 TEST SUMMARY
65 =====
66 General Purpose      : PASSED
67 Audio Processing     : PASSED
68 Image Processing     : PASSED
69 Video Processing     : PASSED
70
71 OVERALL RESULT: 4/4 tests passed
72 ALL TESTS PASSED! Your libraries are working correctly.
73 (multimedia-uv) PS D:\Se\Multimedia> python -u "d:\Se\Multimedia\Tugas1\test_image.py"
74 Generated 2s sine wave at 440Hz
75 Sample rate: 44100Hz
76 Total samples: 88200
77 (multimedia-uv) PS D:\Se\Multimedia>
```

Kode 10: Output verifikasi instalasi

## 4.2 Screenshot Hasil Test

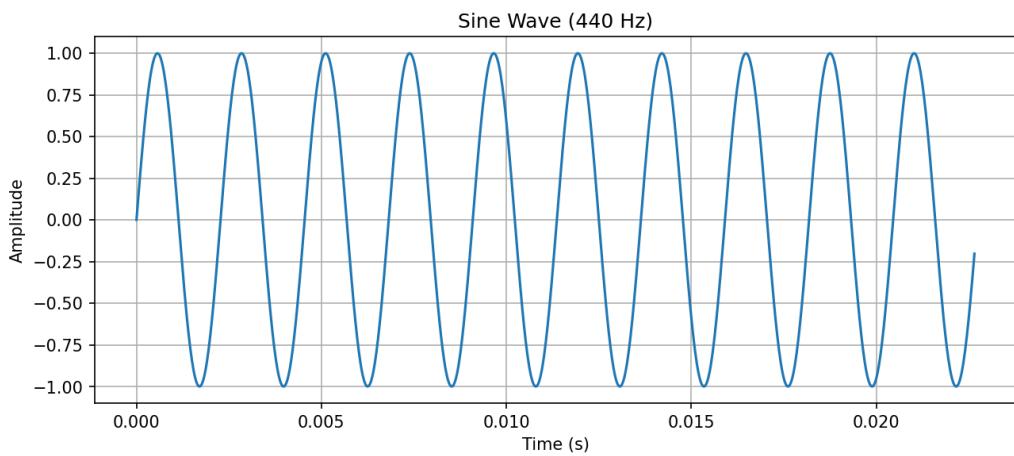
Sisipkan screenshot atau gambar hasil dari:

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

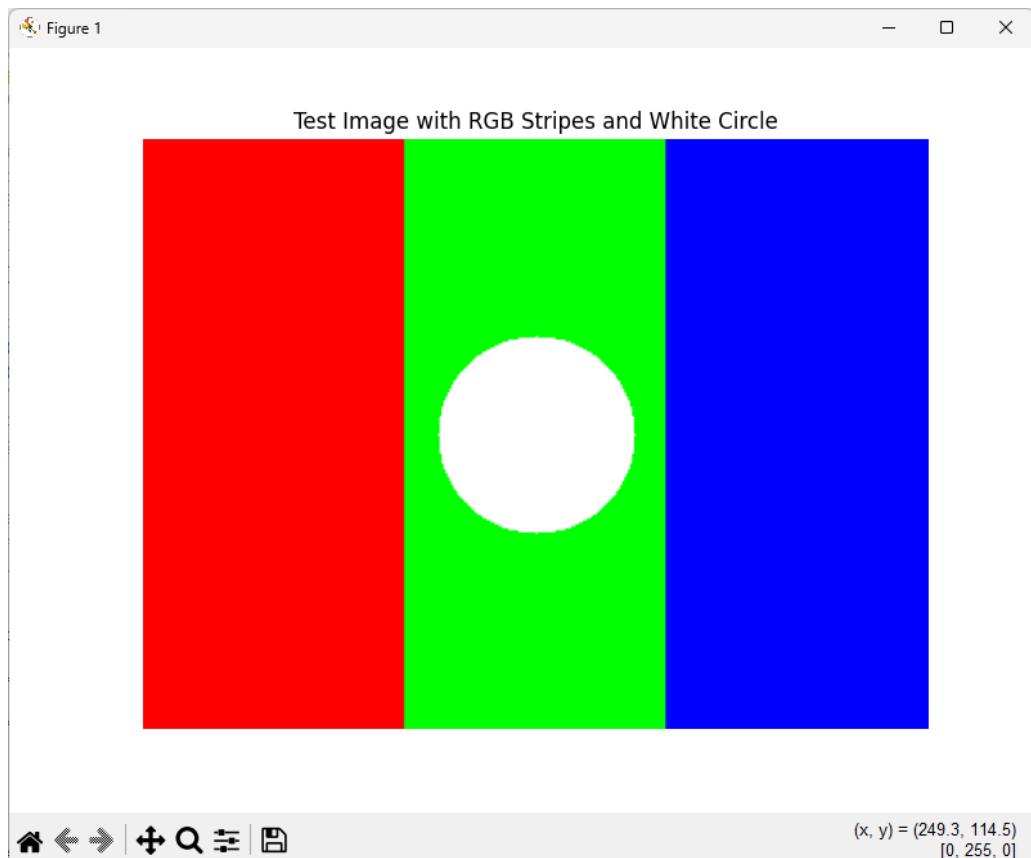
```
● (multimedia-uv) PS D:\Se\Multimedia> where python
● (multimedia-uv) PS D:\Se\Multimedia> Get-Command python

CommandType      Name                           Version   Source
-----          ----                         -----   -----
Application     python.exe                     3.10.18... D:\Se\Multimedia\multimedia-uv\Scripts\python.exe
```

Gambar 5: Tampilan terminal dengan environment aktif



Gambar 6: Output script test audio (sine wave plot)



Gambar 7: Output 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?

*Karena dengan menggunakan environment terpisah, kita dapat mengelola dependensi library yang berbeda untuk setiap project tanpa konflik. Ini juga memudahkan dalam menjaga kondisi lingkungan pengembangan dan memastikan bahwa project dapat digunakan dengan mudah di masa depan.*

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

*Conda adalah manajer paket dan environment yang lebih lengkap, mendukung berbagai bahasa pemrograman dan memiliki banyak paket ilmiah. Venv adalah modul bawaan Python yang sederhana untuk membuat environment virtual, cocok untuk pengguna yang hanya membutuhkan Python. UV adalah alat modern yang cepat dan efisien untuk mengelola environment virtual. Saya memilih UV karena kemudahan penggunaannya dan kecepatan dalam membuat environment serta lebih ringan daripada conda.*

3. Library mana yang paling sulit diinstall dan mengapa?

*moviepy karena memiliki dependensi pada ffmpeg yang kadang sulit diatur pada beberapa sistem operasi.*

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

Dengan menggunakan environment terpisah untuk setiap project, kita dapat menghindari konflik dependensi. Jika terjadi konflik, kita bisa mencoba menginstall versi library yang kompatibel atau menggunakan uv untuk menginstal library yang tidak terinstall dengan baik.

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

*librosa: untuk analisis dan pemrosesan audio. soundfile: untuk membaca dan menulis file audio. scipy: untuk komputasi ilmiah dan teknis. opencv: untuk pemrosesan gambar dan video. pillow: untuk manipulasi gambar. scikit-image: untuk analisis gambar. matplotlib: untuk visualisasi data. ffmpeg: untuk pemrosesan video. moviepy: untuk editing video. numpy: untuk komputasi numerik. pandas: untuk manipulasi data. jupyter: untuk membuat notebook interaktif.*

### 4.4 Troubleshooting

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

- **Masalah 1:** [Referensi tidak muncul padahal sudah menambahkan pada file .bib]

**Solusi:** [menambahkan syntax nocite]

- **Masalah 2:** [Masalah pada saat instalasi library moviepy karena ffmpeg tidak ditemukan]

**Solusi:** [mencoba instalansi berulang dengan command pip install moviepy dan berhasil pada percobaan kedua]

## 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 (multimedia-uv) PS D:\Se\Multimedia> type requirements.txt
2 anyio==4.10.0
3 argon2-cffi==25.1.0
4 argon2-cffi-bindings==25.1.0
5 arrow==1.3.0
6 asttokens==3.0.0
7 async-lru==2.0.5
8 attrs==25.3.0
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==1.3.2
19 cycler==0.12.1
20 debugpy==1.8.16
21 decorator==4.4.2
22 defusedxml==0.7.1
23 exceptiongroup==1.3.0
24 executing==2.2.0
25 fastjsonschema==2.21.2
26 fonttools==4.59.2
27 fqdn==1.5.1
28 h11==0.16.0
29 httpcore==1.0.9
30 httpx==0.28.1
31 idna==3.10
32 imageio==2.37.0
33 imageio-ffmpeg==0.6.0
34 ipykernel==6.30.1
35 ipython==8.37.0
36 ipywidgets==8.1.7
37 isoduration==20.11.0
38 jedi==0.19.2
39 Jinja2==3.1.6
40 joblib==1.5.2
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==2.3.0
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==3.1.4
66 moviepy==1.0.3
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==3.4.2
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 overrides==7.7.0
79 packaging==25.0
80 pandas==2.3.2
81 pandocfilters==1.5.1
82 parso==0.8.5
83 pillow==11.3.0
84 platformdirs==4.4.0
85 pooch==1.8.2
86 proglog==0.1.12
87 prometheus_client==0.22.1
88 prompt_toolkit==3.0.52
89 psutil==7.0.0
90 pure_eval==0.2.3
91 pycparser==2.22
92 Pygments==2.19.2
93 pyparsing==3.2.3
94 python-dateutil==2.9.0.post0
95 python-dotenv==1.1.1
96 python-json-logger==3.3.0
97 pytz==2025.2
98 pywin32==311
99 pywintypes==3.0.0
100 PyYAML==6.0.2
101 pyzmq==27.0.2
102 referencing==0.36.2
103 requests==2.32.5
104 rfc3339-validator==0.1.4
105 rfc3986-validator==0.1.1
106 rfc3987-syntax==1.1.0
107 rpds-py==0.27.1
108 scikit-image==0.25.2
109 scikit-learn==1.7.1
110 scipy==1.15.3
111 Send2Trash==1.8.3
112 six==1.17.0
113 sniffio==1.3.1
114 soundfile==0.13.1
115 soupsieve==2.8
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 tinyccs2==1.4.0
122 toml==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
136 (multimedia-uv) PS D:\Se\Multimedia>
```

Kode 13: Environment/Requirements file

## 6 Kesimpulan

**Tuliskan kesimpulan Anda mengenai:**

- Pengalaman setup Python environment untuk multimedia
  - Setup environment Python untuk multimedia memerlukan perencanaan dan pemahaman yang baik tentang dependency management
  - Penggunaan UV sebagai package manager modern memberikan pengalaman yang lebih cepat dan efisien untuk sekarang ini
  - Proses instalasi library multimedia relatif lancar dengan dokumentasi yang baik
- Persiapan untuk project multimedia selanjutnya
  - Environment yang telah dikonfigurasi dapat digunakan sebagai dasar untuk project-project selanjutnya
  - Requirements.txt yang dihasilkan memudahkan reproduksi environment pada komputer lain
  - Library-library yang terinstall mencakup kebutuhan dasar untuk pemrosesan audio, gambar, dan video
- Saran untuk mahasiswa lain yang akan melakukan setup serupa
  - Pilih package manager yang sesuai dengan kebutuhan dan pengalaman
  - Jangan terpaku dengan 1 cara karena beda device beda cara bahkan satu os bisa memiliki cara berbeda
  - Manfaatkan komunitas dan dokumentasi online untuk mengatasi masalah

## 7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

## References

- [1] Y. Name, “Tutorial belajar LaTeX dasar untuk pemula,” <https://www.sains.web.id/2018/12/tutorial-belajar-latex.html>, 18 Dec. 2018, accessed: 2022-1-6.
- [2] Wikipedia contributors, “LaTeX,” <https://en.wikipedia.org/w/index.php?title=LaTeX&oldid=1060605123>, 16 Dec. 2021, accessed: NA-NA-NA.
- [3] OpenAI. (2023) Chatgpt conversation: Python multimedia environment setup. [Online]. Available: <https://chatgpt.com/share/68b19e52-ca64-8006-ad9d-7f701c2df061>