



---

Nama: **Dinda Joycehana (122140048)** 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  $\text{\LaTeX}$

## 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  $\text{\LaTeX}$  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

```
PS C:\TUGAS KULIAH DINDA\MULTIMEDIA> cd Tugas1
Requirement already satisfied: uv in c:\users\dinda joycehana\appdata\roaming\python\python312\site-packages (0.6.3)

[notice] A new release of pip is available: 25.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\TUGAS KULIAH DINDA\MULTIMEDIA> uv venv --python=python3.10
Using CPython 3.10.16
Creating virtual environment at: .venv
Activate with: .venv\Scripts\activate
PS C:\TUGAS KULIAH DINDA\MULTIMEDIA> .venv\Scripts\activate
(Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA> .venv\Scripts\activate
```

### 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

```
1 # Library Audio Processing
2 (Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1> uv pip install librosa soundfile scipy
3 Resolved 25 packages in 4.08s
4 Prepared 9 packages in 6.97s
5 Installed 25 packages in 4.73s
6 + audioread==3.0.1
7 + certifi==2025.8.3
8 + cffi==1.17.1
9 + charset-normalizer==3.4.3
10 + decorator==5.2.1
11 + idna==3.10
12 + joblib==1.5.2
13 + lazy-loader==0.4
14 + librosa==0.11.0
15 + llvmlite==0.44.0
16 + msgpack==1.1.1
17 + numba==0.61.2
18 + numpy==2.2.6
19 + packaging==25.0
20 + platformdirs==4.4.0
21 + pooch==1.8.2
22 + pycparser==2.22
23 + requests==2.32.5
24 + scikit-learn==1.7.1
25 + scipy==1.15.3
26 + soundfile==0.13.1
27 + soxr==0.5.0.post1
28 + threadpoolctl==3.6.0
29 + typing-extensions==4.15.0
30 + urllib3==2.5.0
31 (Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1>
```

Kode 8: Instalasi library audio

```
1 # Library Image Processing
2 (Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1> uv pip install opencv-python pillow scikit-
  image matplotlib
3 Resolved 18 packages in 2.64s
4 Prepared 8 packages in 34.17s
5 Installed 14 packages in 2.29s
6 + contourpy==1.3.2
7 + cycler==0.12.1
8 + fonttools==4.59.2
9 + imageio==2.37.0
10 + kiwisolver==1.4.9
11 + matplotlib==3.10.5
12 + networkx==3.4.2
13 + opencv-python==4.12.0.88
14 + pillow==11.3.0
15 + pyparsing==3.2.3
16 + python-dateutil==2.9.0.post0
17 + scikit-image==0.25.2
18 + six==1.17.0
19 + tifffile==2025.5.10
```

Kode 9: Instalasi library image

```
1 # Library Video Processing
2 (Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1> uv pip install moviepy
3 Resolved 10 packages in 849ms
4 Prepared 4 packages in 15.09s
5 Installed 6 packages in 259ms
6 + colorama==0.4.6
7 + imageio-ffmpeg==0.6.0
8 + moviepy==2.2.1
9 + proglog==0.1.12
10 + python-dotenv==1.1.1
11 + tqdm==4.67.1
```

Kode 10: Instalasi library video

```
1 # Library General Purpose
2 (Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1> uv pip install numpy pandas jupyter
3 Resolved 105 packages in 39.97s
4 Prepared 31 packages in 26.46s
5 Installed 90 packages in 13.18s
6 + anyio==4.10.0
7 + argon2-cffi==25.1.0
8 + argon2-cffi-bindings==25.1.0
9 + arrow==1.3.0
10 + asttokens==3.0.0
11 + async-lru==2.0.5
12 + attrs==25.3.0
13 + babel==2.17.0
14 + beautifulsoup4==4.13.5
15 + bleach==6.2.0
16 + comm==0.2.3
17 + debugpy==1.8.16
18 + defusedxml==0.7.1
19 + exceptiongroup==1.3.0
20 + executing==2.2.0
21 + fastjsonschema==2.21.2
22 + fqdn==1.5.1
23 + h11==0.16.0
24 + httpcore==1.0.9
25 + httpx==0.28.1
26 + ipykernel==6.30.1
27 + ipython==8.37.0
28 + ipywidgets==8.1.7
29 + isoduration==20.11.0
30 + jedi==0.19.2
31 + jinja2==3.1.6
32 + json5==0.12.1
33 + jsonpointer==3.0.0
34 + jsonschema==4.25.1
35 + jsonschema-specifications==2025.4.1
36 + jupyter==1.1.1
37 + jupyter-client==8.6.3
38 + jupyter-console==6.6.3
39 + jupyter-core==5.8.1
40 + jupyter-events==0.12.0
41 + jupyter-lsp==2.3.0
42 + jupyter-server==2.17.0
43 + jupyter-server-terminals==0.5.3
44 + jupyterlab==4.4.6
45 + jupyterlab-pygments==0.3.0
46 + jupyterlab-server==2.27.3
47 + jupyterlab-widgets==3.0.15
48 + lark==1.2.2
49 + markupsafe==3.0.2
```

```

50 + matplotlib-inline==0.1.7
51 + mistune==3.1.3
52 + nbclient==0.10.2
53 + nbconvert==7.16.6
54 + nbformat==5.10.4
55 + nest-asyncio==1.6.0
56 + notebook==7.4.5
57 + notebook-shim==0.2.4
58 + overrides==7.7.0
59 + pandas==2.3.2
60 + pandocfilters==1.5.1
61 + parso==0.8.5
62 + prometheus-client==0.22.1
63 + prompt-toolkit==3.0.52
64 + psutil==7.0.0
65 + pure-eval==0.2.3
66 + pygments==2.19.2
67 + python-json-logger==3.3.0
68 + pytz==2025.2
69 + pywin32==311
70 + pywinpty==3.0.0
71 + pyyaml==6.0.2
72 + pyzmq==27.0.2
73 + referencing==0.36.2
74 + rfc3339-validator==0.1.4
75 + rfc3986-validator==0.1.1
76 + rfc3987-syntax==1.1.0
77 + rpds-py==0.27.1
78 + send2trash==1.8.3
79 + setuptools==80.9.0
80 + sniffio==1.3.1
81 + soupsieve==2.8
82 + stack-data==0.6.3
83 + terminado==0.18.1
84 + tinycss2==1.4.0
85 + tomli==2.2.1
86 + tornado==6.5.2
87 + traitlets==5.14.3
88 + types-python-dateutil==2.9.0.20250822
89 + tzdata==2025.2
90 + uri-template==1.3.0
91 + wcwidth==0.2.13
92 + webcolors==24.11.1
93 + webencodings==0.5.1
94 + websocket-client==1.8.0
95 + widgetsnbextension==4.0.14
96 (Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1>

```

Kode 11: Instalasi library umum

### 3.4 Bagian 3: Verifikasi Instalasi

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

**Jalankan script dan dokumentasikan hasilnya:**

```

1 # test script to verify installation of various libraries
2 import sys
3 import librosa
4 import soundfile as sf
5 import scipy
6 import cv2
7 from PIL import Image

```

```
8 import skimage
9 import matplotlib.pyplot as plt
10 import moviepy as mpy
11 import numpy as np
12 import pandas as pd
13
14 print("Libraries installed successfully!")
```

Kode 12: Script untuk verifikasi instalasi

### 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 13: 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 14: 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 (Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1> python test_multimedia.py
2 Libraries installed successfully!
3 Generated 2s sine wave at 440Hz
4 Sample rate: 44100Hz
5 Total samples: 88200
6 Created test image: 400x300 pixels
7 Image shape: (300, 400, 3)
8 Image dtype: uint8
9 (Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1>

```

Kode 15: Output verifikasi instalasi

### 4.2 Screenshot Hasil Test

Sisipkan screenshot atau gambar hasil dari:

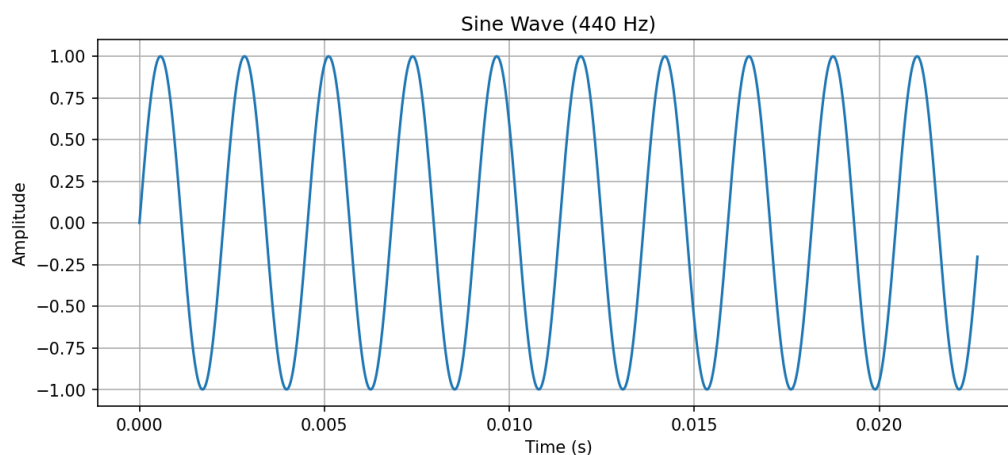
- Terminal/command prompt yang menunjukkan environment aktif
- Output dari script test audio (sine wave plot)
- Output dari script test image (RGB stripes dengan circle)

*Gunakan perintah `\includegraphics` untuk menyisipkan gambar*



```
(Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1> .venv\Scripts\activate
(Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1> python --version
Python 3.10.16
(Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1> python test_multimedia.py
Libraries installed successfully!
Generated 2s sine wave at 440Hz
Sample rate: 44100Hz
Total samples: 88200
Created test image: 400x300 pixels
Image shape: (300, 400, 3)
Image dtype: uint8
(Tugas1) PS C:\TUGAS KULIAH DINDA\MULTIMEDIA\Tugas1> |
```

Gambar 1: Output dari verifikasi environment



Gambar 2: Output dari test audio processing

### 4.3 Analisis dan Refleksi

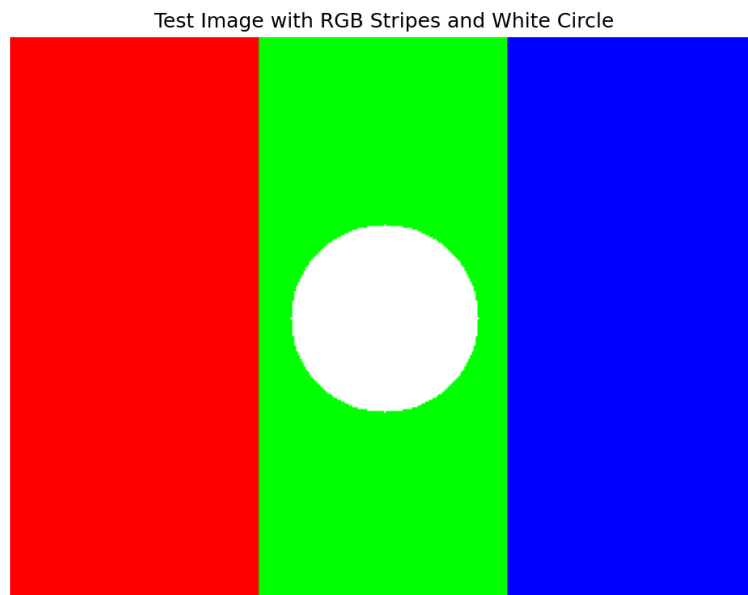
Jawab pertanyaan berikut:

1. Mengapa penting menggunakan environment terpisah untuk project multimedia?

*Karena setiap project multimedia bisa saja memiliki kebutuhan library dan versi yang berbeda-beda. Sehingga dibutuhkan environment terpisah di setiap project agar tidak terjadi konflik antar library.*

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

- Conda adalah package manager sekaligus environment manager yang mendukung berbagai bahasa pemrograman. Karena bisa multilanguages, sehingga populer digunakan dalam data science dan machine learning. Namun conda sizenya lebih besar dan lebih lambat dalam dependency resolution.*
- Sedangkan venv, ini merupakan modul bawaan python dalam membuat virtual environment. Venv lebih ringan dan cepat, namun hanya mendukung package dari python saja.*
- UV mirip dengan venv, hanya saja lebih cepat dan praktis. UV bisa membuat virtual environment sekaligus instalasi package python dengan sangat cepat. uv juga mendukung uv.lock untuk reproducibility. Saya memilih uv karena lebih ringan dan cepat dibanding conda, serta memiliki fitur yang lebih baik dibanding venv.*



Gambar 3: Output dari test image processing

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

*Sejauh ini selama percobaan tidak ada library yang sulit diinstall. Semua library berhasil diinstall dengan lancar.*

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

*Dari saya pribadi, cara mengatasi masalah dependency conflict adalah dengan membuat folder dengan environment baru. Tapi bisa juga dengan menginstal library beserta versinya agar menghindari ketidakcocokan versi. Jika projectnya mirip, mungkin bisa menggunakan requiement.txt yang sama.*

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

(a) *Audio Processing Libraries:*

- *Librosa: library untuk analisis dan pemrosesan audio, seperti ekstraksi fitur, transformasi sinyal, dan visualisasi spektrum.*
- *Soundfile: library untuk membaca dan menulis file audio dalam berbagai format.*
- *Scipy: library untuk komputasi ilmiah. Dipakai untuk filtering, resampling dan transformasi sinyal.*

(b) *Image Processing Libraries:*

- *OpenCV: library untuk pemrosesan gambar dan video, seperti deteksi objek, pengenalan wajah, dan transformasi geometris.*
- *Pillow: library untuk manipulasi gambar, seperti pembacaan, penulisan, dan transformasi gambar.*
- *Scikit-image: library untuk pemrosesan gambar, seperti segmentasi, deteksi tepi, dan transformasi morfologi.*

(c) *Video Processing Libraries:*

- *Movipy: library untuk pemrosesan video, seperti pemotongan, penggabungan, dan penambahan efek pada video.*

(d) *General Purpose Libraries:*

- *Matplotlib*: library untuk membuat grafik dan visualisasi data.
- *Pandas*: library untuk manipulasi dan analisis data, seperti pembacaan dan manipulasi data CSV dan Excel.
- *Numpy*: library untuk komputasi numerik, seperti operasi pada array dan matriks.
- *Jupyter*: library untuk membuat dan berbagi dokumen interaktif yang berisi kode, visualisasi, dan teks.

## 4.4 Troubleshooting

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

- **Masalah:** Tidak bisa mengatur foto dan caption di latex

**Solusi:** Bertanya pada GPT. Ternyata pakai `\begin{figure}`, dan `\caption`, `\label`. Jika mau center maka menggunakan `\centering`. Terkadang fotonya berada di luar margin, jadi harus diatur `widthnya`.

## 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 16: Export conda environment

### 5.2 Untuk venv/uv

```
1 pip freeze > requirements.txt
```

Kode 17: Export pip requirements

Copy-paste isi file `environment.yml` atau `requirements.txt` di sini:

```
1 #isi file requirements.txt (untuk uv)
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==5.2.1
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-client==8.6.3
47 jupyter-console==6.6.3
48 jupyter-core==5.8.1
49 jupyter-events==0.12.0
50 jupyter-lsp==2.3.0
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==3.4.2
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 pywinpty==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 setuptools==80.9.0
113 six==1.17.0
114 sniffio==1.3.1
115 soundfile==0.13.1
116 soupsieve==2.8
117 soxr==0.5.0.post1
118 stack-data==0.6.3
119 terminado==0.18.1
120 threadpoolctl==3.6.0
121 tifffile==2025.5.10
122 tinycss2==1.4.0
123 tomli==2.2.1
124 tornado==6.5.2
125 tqdm==4.67.1
126 traitlets==5.14.3
127 types-python-dateutil==2.9.0.20250822
128 typing-extensions==4.15.0
129 tzdata==2025.2
130 uri-template==1.3.0
131 urllib3==2.5.0
132 wcwidth==0.2.13
133 webcolors==24.11.1
134 webencodings==0.5.1
135 websocket-client==1.8.0
136 widgetsnbextension==4.0.14
```

Kode 18: Environment/Requirements file

## 6 Kesimpulan

Tuliskan kesimpulan Anda mengenai:

- Pengalaman setup Python environment untuk multimedia  
*Tidak ada kesulitan yang berarti selama proses setup. Semua library berhasil diinstall dengan*

*lancar menggunakan uv. Hanya saja, ada beberapa perintah pada modul latihan yang tidak sesuai dengan uv, sehingga saya harus menyesuaikan perintahnya. Contoh pada instalasi library, pada uv seharusnya menggunakan "uv pip install ..." bukan "pip install ...". Kasihan teman-teman yang belum tahu uv, pasti bingung kenapa tidak bisa.*

- Persiapan untuk project multimedia selanjutnya  
*Karena semua dependencies telah didefinisikan pada requirements.txt, maka untuk project multimedia selanjutnya saya hanya perlu membuat environment baru dengan uv, lalu menginstall semua library yang ada di requirements.txt. Sepertinya itu sudah cukup untuk memulai project multimedia.*
- Saran untuk mahasiswa lain yang akan melakukan setup serupa  
*Saran saya pakai uv saja karena lebih ringan dan cepat serta mudah untuk reproducibility. Jangan lupa dokumentasikan semua langkahnya, karena itu penting untuk referensi di masa depan.*

## 7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

## References

[Link ChatGPT Referensi](#)