Nama: Fawwaz Abhitah Sugiarto (122140014) 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 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:

```
# Membuat environment baru dengan nama 'multimedia'
conda create -n multimedia python=3.11

# Mengaktifkan environment
conda activate multimedia

# Verifikasi environment aktif
conda info --envs
```

Kode 1: Membuat environment dengan Conda

## 3.2.2 Opsi 2: Menggunakan venv (Built-in Python)

```
# Membuat environment baru
python3 -m venv multimedia-env

# Mengaktifkan environment (Linux/Mac)
source multimedia-env/bin/activate

# Mengaktifkan environment (Windows)
# multimedia-env\Scripts\activate

# Verifikasi environment aktif
which python
```

Kode 2: Membuat environment dengan venv

## 3.2.3 Opsi 3: Menggunakan uv (Modern dan cepat)

```
# Install uv terlebih dahulu jika belum ada
pip install uv

# Membuat environment baru
uv venv multimedia-uv

# Mengaktifkan environment (Linux/Mac)
source multimedia-uv/bin/activate

# Mengaktifkan environment (Windows)
# multimedia-uv\Scripts\activate

# Verifikasi environment aktif
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

```
O (multimedia-uv) PS C:\Users\okedi\multimedia> uv venv multimedia-uv
Using CPython 3.10.18
Creating virtual environment at: multimedia-uv
? A virtual environment already exists at `multimedia-uv`. Do you want to replace it? [y/n] > yes
hint: Use the `--clear` flag or set `UV_VENV_CLEAR=1` to skip this prompt
```

Gambar 1: Membuat Environment pada Windows

```
    PS C:\Users\okedi\multimedia> multimedia-uv\Scripts\activate
    (multimedia-uv) PS C:\Users\okedi\multimedia> where python
    (multimedia-uv) PS C:\Users\okedi\multimedia>
```

Gambar 2: Mengaktifkan Environment pada Windows

# 3.3 Bagian 2: Instalasi Library Multimedia

Setelah environment aktif, install library-library berikut:

# 3.3.1 Library Audio Processing

```
# Untuk conda:
conda install -c conda-forge librosa soundfile scipy

# Untuk pip (venv/uv):
pip install librosa soundfile scipy
```

Kode 4: Instalasi library audio

## 3.3.2 Library Image Processing

```
# Untuk conda:
conda install -c conda-forge opencv pillow scikit-image matplotlib

# Untuk pip (venv/uv):
pip install opencv-python pillow scikit-image matplotlib
```

Kode 5: Instalasi library image

## 3.3.3 Library Video Processing

```
# Untuk conda:
conda install -c conda-forge ffmpeg
pip install moviepy

# Untuk pip (venv/uv):
pip install moviepy
```

Kode 6: Instalasi library video

## 3.3.4 Library General Purpose

```
# Untuk conda:
conda install numpy pandas jupyter

# Untuk pip (venv/uv):
pip install numpy pandas jupyter
```

Kode 7: Instalasi library umum

#### Dokumentasikan di sini:

- Perintah instalasi yang Anda gunakan
  - 1. pip install librosa soundfile scipy
  - 2. pip install opency-python pillow scikit-image matplotlib
  - 3. pip install moviepy
  - 4. pip install numpy pandas jupyter
- Screenshot proses instalasi atau output sukses

```
(multimedia-uv) PS C:\Users\okedi\multimedia> python -m pip install librosa
Collecting librosa
Using cached librosa-0.11.0-py3-none-any.whl.metadata (8.7 kB)
Collecting audioread>=2.1.9 (from librosa)
Using cached audioread-3.0.1-py3-none-any.whl.metadata (8.4 kB)
Collecting numba>=0.51.0 (from librosa)
Downloading numba-0.61.2-cp310-win_amd64.whl.metadata (2.9 kB)
Collecting numpy>=1.22.3 (from librosa)
Downloading numpy-2.2.6-cp310-cp310-win_amd64.whl.metadata (60 kB)
Collecting scipy>=1.6.0 (from librosa)
Downloading scipy-1.15.3-cp310-cp310-win_amd64.whl.metadata (60 kB)
Collecting scikit-learn>=1.1.0 (from librosa)
Downloading scikit_learn>=1.1.0 (from librosa)
```

Gambar 3: Proses Instalasi Library librosa

```
(multimedia-uv) PS C:\Users\okedi\multimedia> python -m pip install opencv-python pillow scikit-image matplotlib
Collecting opencv-python
Using cached opencv_python-4.12.0.88-cp37-abi3-win_amd64.whl.metadata (19 kB)
Collecting pillow
Downloading pillow-11.3.0-cp310-cp310-win_amd64.whl.metadata (9.2 kB)
Collecting scikit-image
Downloading scikit-image-0.25.2-cp310-cp310-win_amd64.whl.metadata (14 kB)
Collecting matplotlib
Downloading matplotlib-3.10.5-cp310-cp310-win_amd64.whl.metadata (11 kB)
Requirement already satisfied: numpy<2.3.0,>=2 in c:\users\okedi\multimedia-uv\lib\site-packages (from opencv-python) (2.2.6)
Requirement already satisfied: scipy>=1.11.4 in c:\users\okedi\multimedia\multimedia-uv\lib\site-packages (from scikit-image) (1.15.3)
Collecting networkx>=3.0 (from scikit-image) (1.15.3)
Downloading networkx>=3.0 (from scikit-image)
```

Gambar 4: Proses Instalasi Library opency

```
@ (multimedia-nv) PS C:\Users\okedi\multimedia> python -m pip install moviepy
Collecting moviepy
Collecting moviepy
Using cached moviepy-2.2.1-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: decorators6.6,>=4.0.2 in c:\users\okedi\multimedia\multimedia\multimedia\mu\lib\site-packages (from moviepy) (5.2.1)
Requirement already satisfied: imageio.3(a,>=2.5 in c:\users\okedi\multimedia\multimedia\mu\lib\site-packages (from moviepy) (2.37.0)
Collecting imageio.ffmpeg-0.6.2.0 (from moviepy)
Using cached imageio.ffmpeg-0.6.2.0 (from moviepy)
Requirement already satisfied: numpy>=1.25.0 in c:\users\okedi\multimedia\multimedia\multimedia\multimedia-uv\lib\site-packages (from moviepy)
Using cached proglog-0.1.12-py3-none-any.whl.metadata (794 bytes)
Collecting python-dotenv-0.9.10 (from moviepy)
Using cached python-dotenv-1.1.1-py3-none-any.whl.metadata (24 kB)
Requirement already satisfied: pillows(12.0,>=9.2.0 in c:\users\okedi\multimedia\multimedia\multimedia-uv\lib\site-packages (from moviepy) (11.3.0)
Collecting tython-dotenv-1.1.1-py3-none-any.whl.metadata (24 kB)
```

Gambar 5: Proses Instalasi Library moviepy

```
@ (multimedia-uv) PS C:\User\okedi\multimedia> python -m pip install numpy pandas jupyter
Requirement already satisfied: numpy in c:\user\okedi\multimedia-uv\lib\site-packages (2.2.6)
Collecting pandas
Downloading pandas-2.3.2-cp310-cp310-win_amd64.whl.metadata (19 kB)
Collecting jupyter
Using cached jupyter-1.1.1-py2.py3-none-any.whl.metadata (2.0 kB)
Requirement already satisfied: python-dateutil>-2.8.2 in c:\users\okedi\multimedia\multimedia-uv\lib\site-packages (from pandas) (2.9.0.post0)
Collecting pytz>-2020.1 (from pandas)
Using cached pytz-2020.2 (from pandas)
Using cached tzdata-2022.2 (from pandas)
Using cached tzdata-2022.2 (py2.py3-none-any.whl.metadata (2.4 kB)
Collecting tzdata>-2022.7 (from jupyter)
Using cached tzdata-2025.2-py2.py3-none-any.whl.metadata (10 kB)
Collecting jupyter-console (from jupyter)
Using cached notebook.2-d.3-py3-none-any.whl.metadata (10 kB)
Collecting jupyter-console (from jupyter)
Using cached ontebook.2-d.3-py3-none-any.whl.metadata (10 kB)
Collecting jupyter-console (from jupyter)
Using cached jupyter console (5.2-yp2.py3-none-any.whl.metadata (5.8 kB)
```

Gambar 6: Proses Instalasi Library numpy pandas jupyter

• Daftar library yang berhasil diinstall dengan versinya

```
librosa v0.11.0
soundfile v0.13.1
scipy v1.15.3
cv2 (OpenCV) v4.12.0
PIL (Pillow) v11.3.0
scikit-image v0.25.2
matplotlib v3.10.5
moviepy v2.1.2
numpy v2.2.6
pandas v2.3.2
jupyter (notebook) v7.4.5
```

## 3.4 Bagian 3: Verifikasi Instalasi

Buat file Python sederhana untuk menguji semua library yang telah diinstall: Jalankan script dan dokumentasikan hasilnya:

## 3.4.1 Verifikasi Instalasi Library Python

```
print("=== Verifikasi Instalasi Library ===\n")
3 libraries = [
      ("librosa", "import librosa", "librosa.__version__"),
      ("soundfile", "import soundfile", "soundfile.__version__"),
      ("scipy", "import scipy", "scipy.__version__"),
6
      ("cv2 (OpenCV)", "import cv2", "cv2.__version__"),
      ("PIL (Pillow)", "from PIL import Image, __version_ as pil_version", "pil_version"),
      ("scikit-image", "import skimage", "skimage.__version__"),
9
      ("matplotlib", "import matplotlib", "matplotlib.__version__"),
10
      ("moviepy", "import moviepy", "moviepy.__version__"),
11
      ("numpy", "import numpy", "numpy.__version__"),
12
      ("pandas", "import pandas", "pandas.__version__"),
13
      ("jupyter (notebook)", "import notebook", "notebook.__version__"),
14
15
16
17 for name, statement, version_attr in libraries:
      try:
18
          exec(statement)
19
          try:
20
              version = eval(version_attr)
21
              print(f"[OK] {name} terinstall (versi: {version})")
22
          except Exception:
              print(f"[OK] {name} terinstall (versi tidak dapat dideteksi)")
25
      except Exception as e:
          print(f"[FAIL] {name} gagal di-import -> {e}")
26
28 print("\n=== Verifikasi selesai ===")
```

Kode 8: Verifikasi instalasi library multimedia Python

```
(multimedia-uv) PS C:\Users\okedi\multimedia> Python verifikasi_installasi.py
=== Verifikasi Instalasi Library ===

[OK] librosa terinstall (versi: 0.11.0)
[OK] soundfile terinstall (versi: 0.13.1)
[OK] scipy terinstall (versi: 1.15.3)
[OK] cv2 (OpenCV) terinstall (versi: 4.12.0)
[OK] PIL (Pillow) terinstall (versi: 11.3.0)
[OK] scikit-image terinstall (versi: 0.25.2)
[OK] matplotlib terinstall (versi: 3.10.5)
[OK] moviepy terinstall (versi: 2.1.2)
[OK] numpy terinstall (versi: 2.3.2)
[OK] pandas terinstall (versi: 2.3.2)
[OK] jupyter (notebook) terinstall (versi: 7.4.5)
=== Verifikasi selesai ===
```

Gambar 7: Output Verifikasi Instalasi Library Multimedia

## 3.5 Bagian 4: Simple Test dengan Sample Code

Buat dan jalankan contoh sederhana untuk setiap kategori multimedia:

#### 3.5.1 Test Audio Processing

```
import numpy as np
import matplotlib.pyplot as plt
4 # Generate simple sine wave
5 duration = 2 # seconds
6 sample_rate = 44100
7 frequency = 440 # A4 note
9 t = np.linspace(0, duration, int(sample_rate * duration))
audio_signal = np.sin(2 * np.pi * frequency * t)
11
12 # Plot waveform
plt.figure(figsize=(10, 4))
plt.plot(t[:1000], audio_signal[:1000]) # Plot first 1000 samples
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()
22 print(f"Generated {duration}s sine wave at {frequency}Hz")
print(f"Sample rate: {sample_rate}Hz")
24 print(f"Total samples: {len(audio_signal)}")
```

Kode 9: Test audio processing sederhana

## 3.5.2 Test Image Processing

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image

# Create a simple test image
width, height = 400, 300
image = np.zeros((height, width, 3), dtype=np.uint8)
```

```
9 # Add some patterns
image[:, :width//3, 0] = 255 # Red section
image[:, width//3:2*width//3, 1] = 255 # Green section
image[:, 2*width//3:, 2] = 255 # Blue section
14 # Add a white circle in the center
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</pre>
image[mask] = [255, 255, 255]
21 # Display and save
plt.figure(figsize=(8, 6))
plt.imshow(image)
24 plt.title('Test Image with RGB Stripes and White Circle')
plt.axis('off')
26 plt.savefig('test_image.png', dpi=150, bbox_inches='tight')
27 plt.show()
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 10: Test image processing sederhana

#### Dokumentasikan hasil eksekusi:

• Screenshot output dari kedua script di atas

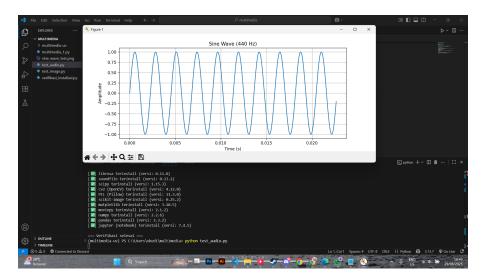
```
    (multimedia-uv) PS C:\Users\okedi\multimedia> python test_audio.py
        Generated 2s sine wave at 440Hz
        Sample rate: 44100Hz
        Total samples: 88200
```

Gambar 8: Output dari script test audio

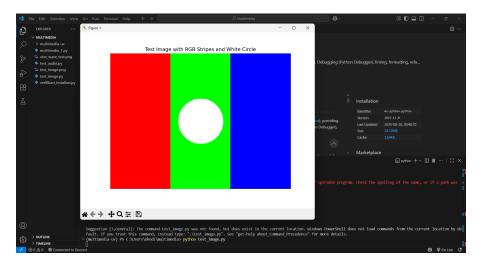
```
    (multimedia-uv) PS C:\Users\okedi\multimedia> python test_image.py
    Created test image: 400x300 pixels
    Image shape: (300, 400, 3)
    Image dtype: uint8
```

Gambar 9: Output dari script test image

• Gambar yang dihasilkan (sine\_wave\_test.png dan test\_image.png)



Gambar 10: Hasil dari script test audio



Gambar 11: Hasil dari script test image RGB

• Error message jika ada dan cara mengatasinya

# 4 Bagian Laporan

## 4.1 Output Verifikasi Instalasi

Copy-paste output lengkap dari script verifikasi\_instalasi.py di sini:

```
1 === Verifikasi Instalasi Library ===
2
3 [OK] librosa terinstall (versi: 0.8.1)
4 [OK] soundfile terinstall (versi: 0.10.3)
5 [OK] scipy terinstall (versi: 1.7.1)
6 [OK] cv2 (OpenCV) terinstall (versi: 4.5.3)
7 [OK] PIL (Pillow) terinstall (versi: 8.4.0)
8 [OK] scikit-image terinstall (versi: 0.18.3)
9 [OK] matplotlib terinstall (versi: 3.4.3)
10 [OK] moviepy terinstall (versi: 1.0.3)
11 [OK] numpy terinstall (versi: 1.21.2)
12 [OK] pandas terinstall (versi: 1.3.3)
```

Worksheet 1: Setup Python Environment untuk Multimedia

```
13 [OK] jupyter (notebook) terinstall (versi: 1.0.0)
14
15 === Verifikasi selesai ===
```

Kode 11: Output verifikasi instalasi

## 4.2 Screenshot Hasil Test

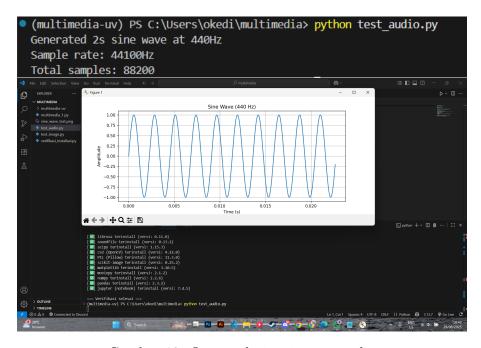
## Sisipkan screenshot atau gambar hasil dari:

• Terminal/command prompt yang menunjukkan environment aktif

```
(multimedia-uv) PS C:\Users\okedi\multimedia> uv venv multimedia-uv
Using CPython 3.10.18
Creating virtual environment at: multimedia-uv
? A virtual environment already exists at `multimedia-uv`. Do you want to replace it? [y/n] > yes
hint: Use the `--clear` flag or set `UV_VENV_CLEAR=1` to skip this prompt
```

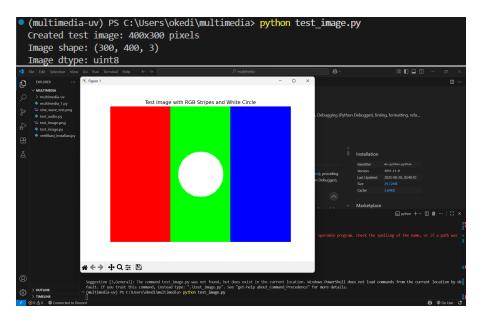
Gambar 12: Environment aktif di terminal

• Output dari script test audio (sine wave plot)



Gambar 13: Output dari script test audio

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



Gambar 14: Output dari script test image

#### 4.3 Analisis dan Refleksi

# Jawab pertanyaan berikut:

## 1. Mengapa penting menggunakan environment terpisah untuk project multimedia?

Penting untuk menggunakan environment terpisah untuk setiap proyek multimedia. Hal ini dikarenakan masing-masing proyek biasanya membutuhkan library dan versi yang berbeda. Sebagai contoh, pada proyek 1 membutuhkan versi tertentu dari sebuah library, sementara proyek 2 memerlukan versi yang berbeda. Jika kedua proyek dijalankan dalam environment yang sama, versi library akan saling bertabrakan. Dengan membuat environment terpisah, setiap proyek bisa menggunakan versi library yang sesuai tanpa menimbulkan konflik atau error.

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

Menurut saya perbedaan antar conda, venv, dan uv terletak pada masing-masing fiturnya conda walaupun toolnya yang termasuk berat tetapi conda memiliki banyak paket yang sudah terinstall dan mudah digunakan untuk proyek multimedia, sedangkan venv adalah tool bawaan python yang ringan dan sederhana, namun memerlukan instalasi manual untuk setiap library. Sedangkan uv adalah tool yang lebih modern dan cepat dibandingkan venv, serta memiliki fitur tambahan seperti manajemen dependency yang lebih baik. Saya memakai uv dikarenakan uv lebih ringan dan cepat.

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

tadinya kesulitan install library dikarenakan saat melakukan instalasi saya salah memilih environment yang digunakan. tetapi sebenarnya dalam melakukan instalasi library itu sangat mudah

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

Apabila ada library yang mengalami masalah versi, melakukan upgrade akan membuat UV secara otomatis memasang versi yang sesuai. Selain itu, perintah uv pip check dapat digunakan untuk melihat library mana yang memiliki konflik versi, sehingga bisa diperbaiki dengan menghapus atau memperbaruinya.

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

- librosa: Analisis audio/musik dan ekstraksi data dari file audio.
- soundfile: Membaca dan menulis file audio.
- scipy: Algoritma untuk optimisasi, aljabar linear, FFT, dan statistik.
- cv2 (OpenCV): Computer vision untuk memproses dan menganalisis gambar/video.
- PIL (Pillow): Manipulasi gambar, seperti resize, crop, filter, dan konversi format.
- scikit-image: Pengolahan citra, termasuk filter dan transformasi.
- matplotlib: Membuat grafik, plot, dan visualisasi data.
- moviepy: Mengedit dan memproses video.
- numpy: Komputasi numerik dengan array/matriks.
- pandas: Analisis dan manipulasi data tabel.
- jupyter (notebook): Notebook interaktif untuk kode, teks, grafik, dan eksekusi.

# 4.4 Troubleshooting

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

- Masalah 1: Penggunaan Which yang salah, seharusnya menggunakan Where. Saat saya menjalankan perintah, saya mendapatkan error yang menyatakan bahwa 'Which' tidak ditemukan.
  - Solusi: Setelah saya bertanya melalui GPT, GPT sendiri menyuruh untuk menggunakan where yang dimana where tersebut merupakan perintah yang digunakan untuk windows sedangkan untuk where merupakan perintah yang ada di Linux/Macbook. Saya memperbaiki perintah dengan mengganti 'Which' menjadi 'Where' dan menjalankannya kembali.
- Masalah 2: Salah dalam melakukan instalasi library

Solusi: Ketika saya bertanya kepada GPT ternyata pip yang berada pada multimedia-ev tidak ada, kemudian saya melakukan penginstalan pip secara manual, lalu saya menjalankan perintah menggunakan -m supaya agar menjalakan module sebagai script.

# 5 Export Environment untuk Reproduksi

Sebagai langkah terakhir, export environment Anda agar dapat direproduksi:

## 5.1 Untuk Conda

conda env export > environment.yml

Kode 12: Export conda environment

# 5.2 Untuk venv/uv

pip freeze > requirements.txt

Kode 13: Export pip requirements

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

```
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
beautifulsoup4==4.13.5
11 bleach==6.2.0
12 certifi==2025.8.3
13 cffi==1.17.1
charset-normalizer==3.4.3
15 colorama==0.4.6
16 \text{ comm} = 0.2.3
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
14 fastjsonschema==2.21.2
25 fonttools==4.59.2
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 \text{ imageio} = 2.37.0
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==1.5.2
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-console==6.6.3
46 jupyter-events==0.12.0
jupyter-lsp==2.3.0
48 jupyter_client==8.6.3
49 jupyter_core==5.8.1
50 jupyter_server==2.17.0
51 jupyter_server_terminals==0.5.3
52 jupyterlab==4.4.6
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==3.1.4
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==4.4.0
84 pooch==1.8.2
85 proglog==0.1.12
86 prometheus_client==0.22.1
87 prompt_toolkit==3.0.52
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
referencing==0.36.2
102 requests==2.32.5
103 rfc3339-validator==0.1.4
rfc3986-validator==0.1.1
105 rfc3987-syntax==1.1.0
106 rpds-py==0.27.1
107 scikit-image==0.25.2
108 scikit-learn==1.7.1
109 scipy==1.15.3
110 Send2Trash==1.8.3
six = 1.17.0
112 sniffio==1.3.1
113 soundfile==0.13.1
114 soupsieve==2.8
115 soxr==0.5.0.post1
stack-data==0.6.3
117 terminado==0.18.1
threadpoolctl==3.6.0
119 tifffile==2025.5.10
120 tinycss2==1.4.0
121 tomli==2.2.1
122 tornado==6.5.2
123 tqdm==4.67.1
124 traitlets==5.14.3
```

```
types-python-dateutil==2.9.0.20250822
typing_extensions==4.15.0
tzdata==2025.2
uri-template==1.3.0
urllib3==2.5.0
wcwidth==0.2.13
webcolors==24.11.1
webencodings==0.5.1
websocket-client==1.8.0
widgetsnbextension==4.0.14
```

Kode 14: 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

Pengalaman saya dalam melakukan setup python environment adalah lumayan cukup pusing, terutama dalam mengatasi permasalah-permasalahan tadi. Namun, dengan bantuan dokumentasi dan forum online, saya berhasil menyelesaikan masalah yang ada. Untuk project multimedia selanjutnya, saya akan lebih berhati-hati dalam dalam mengikuti langkah-langkah dan memperhatikan dependensi yang digunakan dan mencoba untuk menggunakan virtual environment agar lebih terisolasi. Saran saya untuk mahasiswa lain yang akan melakukan setup serupa adalah untuk selalu berhati-hati dalam mengikuti langkah-langkah yang ada sehingga tidak terjadi kesalahan yang bisa mempersulit proses setup.

# 7 Referensi

Referensi Troubleshooting Chat GPT - Open AI

| Fawwaz Abhitah Sugiarto (122140014) | 15 |
|-------------------------------------|----|
| References                          |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |
|                                     |    |