Program Studi Teknik Informatika
Institut Teknologi Sumatera

Nama: **Rachel Olivia Manullang (122140181)** 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

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

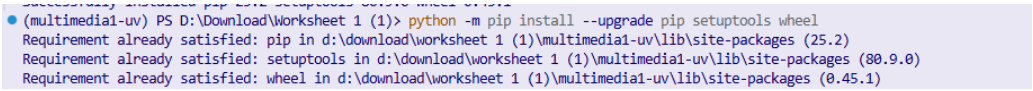
```
# Untuk conda:
conda install numpy pandas jupyter

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

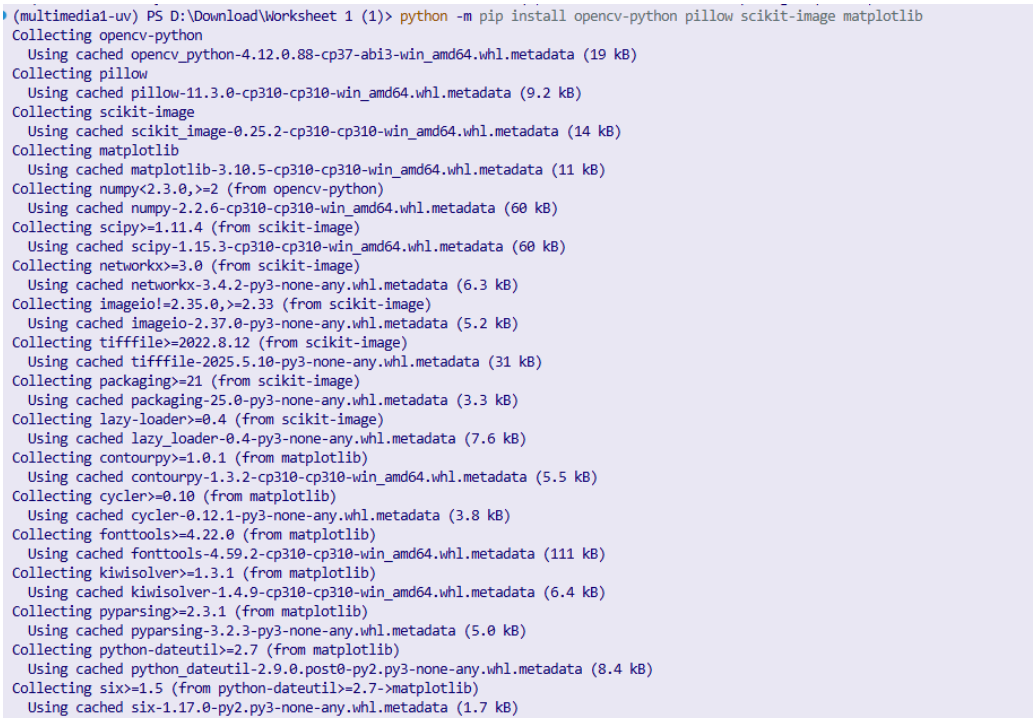
Kode 7: Instalasi library umum

Dokumentasikan di sini:

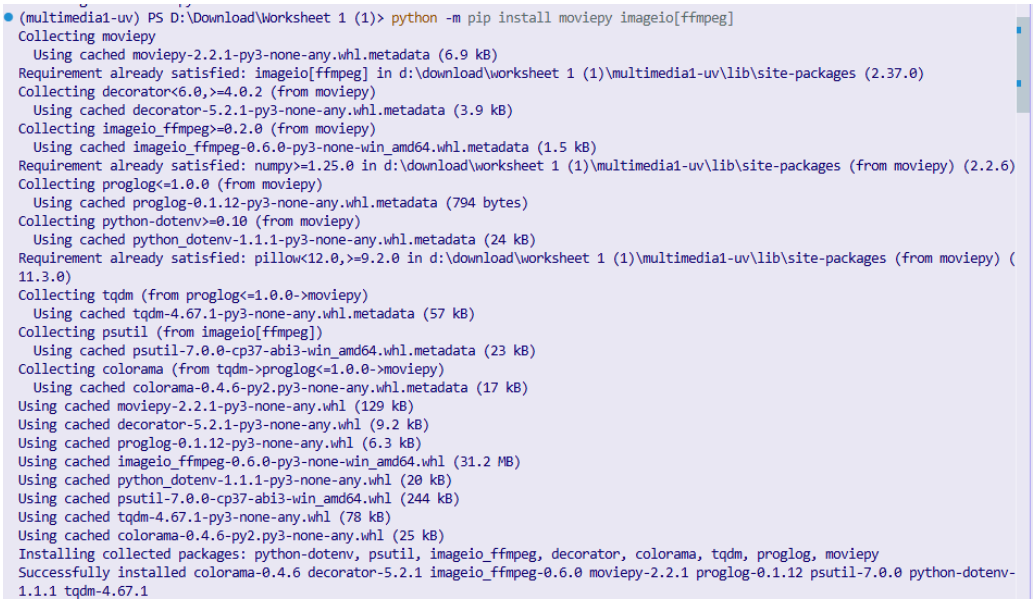
- Perintah instalasi yang Anda gunakan
- Screenshot proses instalasi atau output sukses
- Daftar library yang berhasil diinstall dengan versinya



Gambar 1: Library Audio Processing berhasil diinstall



Gambar 2: Library Image Processing berhasil diinstall



Gambar 3: Library Video Processing berhasil diinstall

```
(multimedia1-uv) PS D:\Download\Worksheet 1 (1)> python -m pip install numpy pandas jupyter
Requirement already satisfied: numpy in d:\download\worksheet 1 (1)\multimedia1-uv\lib\site-packages (2.2.6)
Collecting pandas
  Using cached 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 d:\download\worksheet 1 (1)\multimedia1-uv\lib\site-packages (from pandas)
 (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Using cached pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Using cached tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting notebook (from jupyter)
  Using cached notebook-7.4.5-py3-none-any.whl.metadata (10 kB)
Collecting jupyter-console (from jupyter)
  Using cached jupyter_console-6.6.3-py3-none-any.whl.metadata (5.8 kB)
Collecting nbconvert (from jupyter)
  Using cached nbconvert-7.16.6-py3-none-any.whl.metadata (8.5 kB)
Collecting ipykernel (from jupyter)
  Using cached ipykernel-6.30.1-py3-none-any.whl.metadata (6.2 kB)
Collecting ipywidgets (from jupyter)
  Using cached ipywidgets-8.1.7-py3-none-any.whl.metadata (2.4 kB)
Collecting jupyterlab (from jupyter)
  Using cached jupyterlab-4.4.6-py3-none-any.whl.metadata (16 kB)
Requirement already satisfied: six>=1.5 in d:\download\worksheet 1 (1)\multimedia1-uv\lib\site-packages (from python-dateutil>=2.8.
2->pandas) (1.17.0)
Collecting comm>=0.1.1 (from ipykernel->jupyter)
  Using cached comm-0.2.3-py3-none-any.whl.metadata (3.7 kB)
Collecting debugpy>=1.6.5 (from ipykernel->jupyter)
  Using cached debugpy-1.8.16-cp310-cp310-win_amd64.whl.metadata (1.4 kB)
Collecting ipython>=7.23.1 (from ipykernel->jupyter)
  Using cached ipython-8.37.0-py3-none-any.whl.metadata (5.1 kB)
Collecting jupyter-client>=8.0.0 (from ipykernel->jupyter)
  Using cached jupyter_client-8.6.3-py3-none-any.whl.metadata (8.3 kB)
Collecting jupyter-core!=5.0.*,>=4.12 (from ipykernel->jupyter)
  Using cached jupyter_core-5.8.1-py3-none-any.whl.metadata (1.6 kB)
Collecting matplotlib-inline>=0.1 (from ipykernel->jupyter)
  Using cached matplotlib_inline-0.1.7-py3-none-any.whl.metadata (3.9 kB)
Collecting nest-asyncio>=1.4 (from ipykernel->jupyter)
  Using cached nest_asyncio-1.6.0-py3-none-any.whl.metadata (2.8 kB)
Requirement already satisfied: packaging>=22 in d:\download\worksheet 1 (1)\multimedia1-uv\lib\site-packages (from ipykernel->jupyt
er) (25.0)
Requirement already satisfied: psutil>=5.7 in d:\download\worksheet 1 (1)\multimedia1-uv\lib\site-packages (from ipykernel->jupyter
) (7.0.0)
Collecting pyzmq>=25 (from ipykernel->jupyter)
  Using cached pyzmq-27.0.2-cp310-cp310-win_amd64.whl.metadata (6.0 kB)
Collecting tornado>=6.2 (from ipykernel->jupyter)
  Using cached tornado-6.5.2-cp39-abi3-win_amd64.whl.metadata (2.9 kB)
Collecting traitlets>=5.4.0 (from ipykernel->jupyter)
  Using cached traitlets-5.14.3-py3-none-any.whl.metadata (10 kB)
```

Gambar 4: Library General Purpose berhasil diinstall

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

```python
import numpy as np
import matplotlib.pyplot as plt

# Generate simple sine wave
duration = 2  # seconds
sample_rate = 44100
frequency = 440  # A4 note

t = np.linspace(0, duration, int(sample_rate * duration))
audio_signal = np.sin(2 * np.pi * frequency * t)

# Plot waveform
plt.figure(figsize=(10, 4))
plt.plot(t[:1000], audio_signal[:1000])  # Plot first 1000 samples
```

```
plt.title('Sine Wave (440 Hz)')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.grid(True)
plt.savefig('sine_wave_test.png', dpi=150, bbox_inches='tight')
plt.show()

print(f"Generated {duration}s sine wave at {frequency}Hz")
print(f"Sample rate: {sample_rate}Hz")
print(f"Total samples: {len(audio_signal)}")
```

Kode 8: 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)

# 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

# Add a white circle in the center
center_x, center_y = width//2, height//2
radius = 50
Y, X = np.ogrid[:height, :width]
mask = (X - center_x)**2 + (Y - center_y)**2 <= radius**2
image[mask] = [255, 255, 255]

# Display and save
plt.figure(figsize=(8, 6))
plt.imshow(image)
plt.title('Test Image with RGB Stripes and White Circle')
plt.axis('off')
plt.savefig('test_image.png', dpi=150, bbox_inches='tight')
plt.show()

print(f"Created test image: {width}x{height} pixels")
print(f"Image shape: {image.shape}")
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:

```
(multimedia1-uv) PS D:\Download\Worksheet 1 (1)> python -u "d:\Download\Worksheet 1 (1)\Tugas1\
    validasi.py"
=== Test Library Audio ===
[OK] Audio libs terimport
   Librosa, soundfile, scipy OK (contoh sinyal sinus dibuat)

=== Test Library Image ===
[OK] Image libs terimport
   OpenCV, Pillow, scikit-image, matplotlib OK (gambar dummy dibuat)

=== Test Library Video ===
[OK] Video libs terimport
   MoviePy OK (video clip dummy dibuat)

=== Test Library General Purpose ===
[OK] General purpose libs terimport
   Numpy, Pandas OK (dataframe dibuat):
   A  B
0  1  4
1  2  5
2  3  6

Selesai.
(multimedia1-uv) PS D:\Download\Worksheet 1 (1)>
```

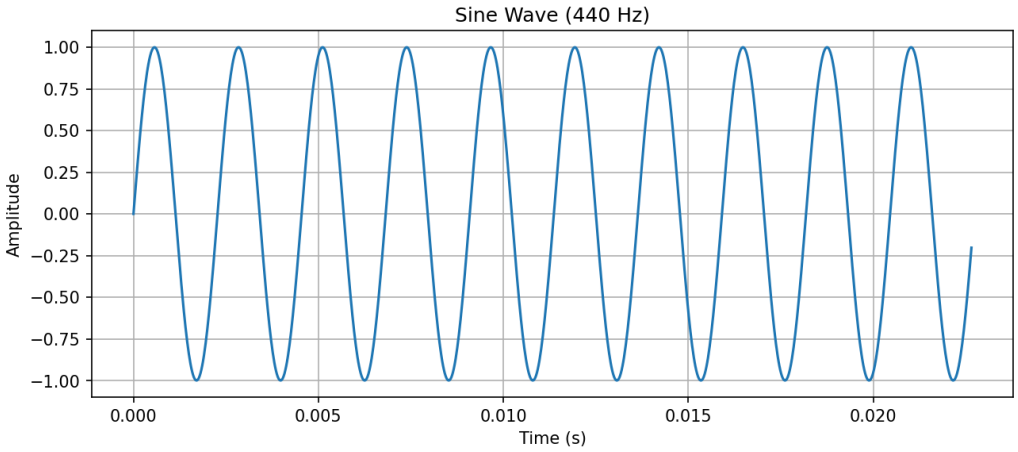
Kode 10: Output verifikasi instalasi

4.2 Screenshot Hasil Test

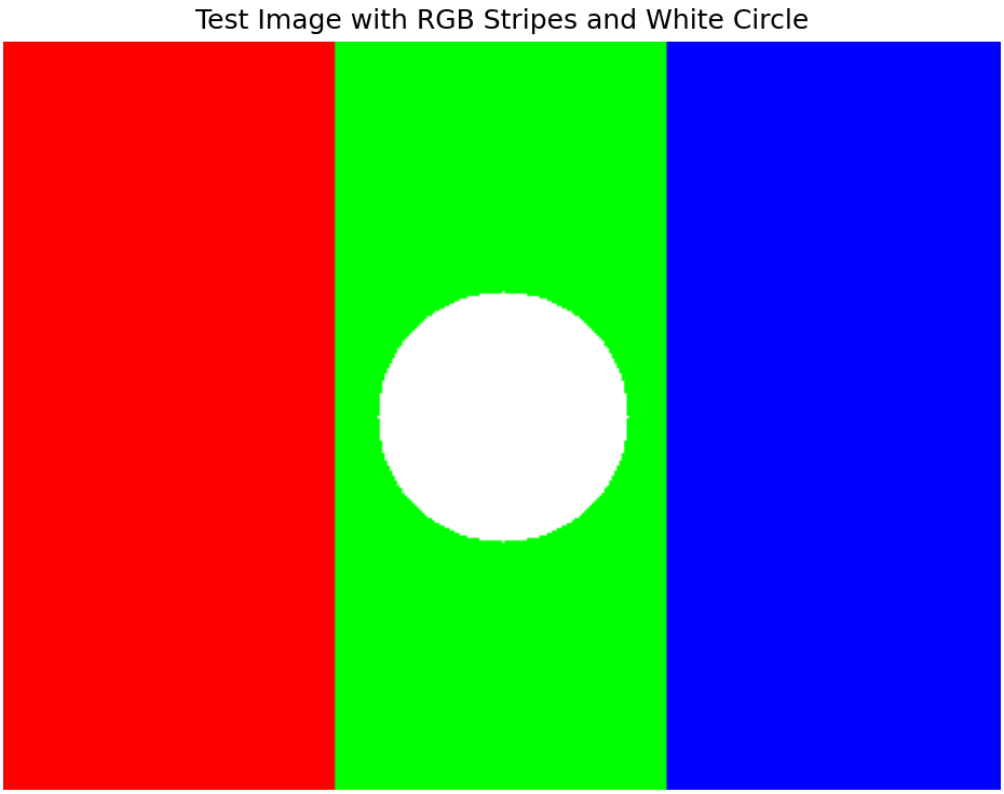
Sisipkan screenshot atau gambar hasil dari:

- 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 5: Tampilan environment aktif



Gambar 6: Output script test audio



Gambar 7: Output script test image

Gunakan perintah `\includegraphics` *untuk menyisipkan gambar*

4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

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

   [Environment terpisah penting supaya library multimedia yang berat dan beda versi nggak bentrok dengan project lain.]

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

   [venv simpel bawaan Python, conda bisa atur dependency non-Python (kayak CUDA), sedangkan uv lebih modern dan cepat. Aku pilih conda karena lebih aman buat multimedia.]

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

   [Paling susah biasanya MoviePy, karena banyak bergantung pada FFmpeg dan codec video yang kadang harus diinstal manual.]

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

   [Kalau ada conflict, biasanya bikin environment baru atau pakai file requirements.txt/environment.yml biar konsisten.]

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

   [Contoh fungsi library: OpenCV (olah gambar/video), NumPy (perhitungan array), Matplotlib (grafik), PyTorch (deep learning), Pillow (edit gambar).]

4.4 Troubleshooting

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

- **Masalah 1:** *[Deskripsi masalah]*

  Solusi: *[Cara mengatasi]*

- **Masalah 2:** *[Deskripsi masalah]*

  Solusi: *[Cara mengatasi]*

5 Export Environment untuk Reproduksi

Sebagai langkah terakhir, export environment Anda agar dapat direproduksi:

5.1 Untuk Conda

```
conda env export > environment.yml
```

Kode 11: Export conda environment

5.2 Untuk venv/uv

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

Kode 12: Export pip requirements

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

```
(multimedia1-uv) PS D:\Download\Worksheet 1 (1)> type requirements.txt
anyio==4.10.0
argon2-cffi==25.1.0
argon2-cffi-bindings==25.1.0
arrow==1.3.0
asttokens==3.0.0
async-lru==2.0.5
attrs==25.3.0
audioread==3.0.1
```

```
babel==2.17.0
beautifulsoup4==4.13.5
bleach==6.2.0
certifi==2025.8.3
cffi==1.17.1
charset-normalizer==3.4.3
colorama==0.4.6
comm==0.2.3
contourpy==1.3.2
cycler==0.12.1
debugpy==1.8.16
decorator==4.4.2
defusedxml==0.7.1
exceptiongroup==1.3.0
executing==2.2.0
fastjsonschema==2.21.2
fonttools==4.59.2
fqdn==1.5.1
h11==0.16.0
httpcore==1.0.9
httpx==0.28.1
idna==3.10
imageio==2.37.0
imageio-ffmpeg==0.6.0
ipykernel==6.30.1
ipython==8.37.0
ipywidgets==8.1.7
isoduration==20.11.0
jedi==0.19.2
Jinja2==3.1.6
joblib==1.5.2
json5==0.12.1
jsonpointer==3.0.0
jsonschema==4.25.1
jsonschema-specifications==2025.4.1
jupyter==1.1.1
jupyter-console==6.6.3
jupyter-events==0.12.0
jupyter-lsp==2.3.0
jupyter_client==8.6.3
jupyter_core==5.8.1
jupyter_server==2.17.0
jupyter_server_terminals==0.5.3
jupyterlab==4.4.6
jupyterlab_pygments==0.3.0
jupyterlab_server==2.27.3
jupyterlab_widgets==3.0.15
kiwisolver==1.4.9
lark==1.2.2
lazy_loader==0.4
librosa==0.11.0
llvmlite==0.44.0
MarkupSafe==3.0.2
matplotlib==3.10.5
matplotlib-inline==0.1.7
mistune==3.1.4
moviepy==1.0.3
msgpack==1.1.1
nbclient==0.10.2
nbconvert==7.16.6
nbformat==5.10.4
nest-asyncio==1.6.0
```

```
networkx==3.4.2
notebook==7.4.5
notebook_shim==0.2.4
numba==0.61.2
numpy==2.2.6
opencv-python==4.12.0.88
overrides==7.7.0
packaging==25.0
pandas==2.3.2
pandocfilters==1.5.1
parso==0.8.5
pillow==11.3.0
platformdirs==4.4.0
pooch==1.8.2
proglog==0.1.12
prometheus_client==0.22.1
prompt_toolkit==3.0.52
psutil==7.0.0
pure_eval==0.2.3
pycparser==2.22
Pygments==2.19.2
pyparsing==3.2.3
python-dateutil==2.9.0.post0
python-dotenv==1.1.1
python-json-logger==3.3.0
pytz==2025.2
pywin32==311
pywinpty==3.0.0
PyYAML==6.0.2
pyzmq==27.0.2
referencing==0.36.2
requests==2.32.5
rfc3339-validator==0.1.4
rfc3986-validator==0.1.1
rfc3987-syntax==1.1.0
rpds-py==0.27.1
scikit-image==0.25.2
scikit-learn==1.7.1
scipy==1.15.3
Send2Trash==1.8.3
six==1.17.0
sniffio==1.3.1
soundfile==0.13.1
soupsieve==2.8
soxr==0.5.0.post1
stack-data==0.6.3
terminado==0.18.1
threadpoolctl==3.6.0
tifffile==2025.5.10
tinycss2==1.4.0
tomli==2.2.1
tornado==6.5.2
tqdm==4.67.1
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
(multimedia1-uv) PS D:\Download\Worksheet 1 (1)>
```

Kode 13: Environment/Requirements file

6 Kesimpulan

Tuliskan kesimpulan Anda mengenai:

- Pengalaman setup Python environment untuk multimedia
  - Awalnya agak ribet karena beberapa library butuh dependency eksternal, tapi setelah pakai environment terpisah proses jadi lebih terkontrol.
- Persiapan untuk project multimedia selanjutnya
  - Aku akan simpan file requirement.txt supaya bisa cepat replikasi environment di laptop lain.
- Saran untuk mahasiswa lain yang akan melakukan setup serupa
  - Kalau pakai uv, manfaatkan kecepatannya untuk install ulang environment dari awal dan biasakan cek dokumentasi library kalau ada error dependency.

[Kesimpulan Anda di sini]

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] Python Packaging Authority, *Installing Using Pip and Virtual Environments*, 2023, accessed: 2025-08-29. [Online]. Available: https://packaging.python.org/guides/installing-using-pip-and-virtual-environments/

[4] Astral, *uv Documentation*, 2025, accessed: 2025-08-29. [Online]. Available: https://docs.astral.sh/uv/

[5] Zulko, *MoviePy Installation Guide*, 2023, accessed: 2025-08-29. [Online]. Available: https://zulko.github.io/moviepy/install.html

[6] OpenCV Team, *OpenCV Documentation*, 2023, accessed: 2025-08-29. [Online]. Available: https://docs.opencv.org/