Program Studi Teknik Informatika
Institut Teknologi Sumatera

Nama: **Fathan Andi Kartagama (122140055)** Tugas Ke: **Worksheet 1: Setup Python Environment untuk Multimedia**

Mata Kuliah: **Sistem Teknologi Multimedia (IF25-40305)** Tanggal: August 26, 2025

1 Instruksi Tugas

1.1 Persiapan

- Menginstall Python 3.8 atau lebih baru (disarankan 3.10)
- Memilih salah satu tool manajemen environment: **conda**, **venv**, atau **uv**. Pada mata kuliah ini saya menggunakan manajemen environtment UV
- Membuka terminal/command prompt
- Menyiapkan dokumen LaTeX ini untuk dokumentasi

1.2 Bagian 1: Membuat Environment Python

1.2.1 Menggunakan uv

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

# bisa dengan powershell (Windows)
powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"

# atau terminal (MacOS dan Linux)
curl -LsSf https://astral.sh/uv/install.sh | sh

# 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 (MacOS dan Linux)
which python

# Verifikasi environtment aktif (Windows)
## Jika Menggunakan CMD
where python

## Jika Menggunakan Powershell
Get-Command python | Select-Object -ExpandProperty Source
```

Kode 1: Membuat environment dengan uv

Dokumentasikan di sini:

- Tool manajemen environment yang Anda pilih: **[UV]**
- Screenshot atau copy-paste output dari perintah verifikasi environment

Gambar 1: Output verifikasi environment

1.3 Bagian 2: Instalasi Library Multimedia

Setelah environment aktif, install library-library berikut:

1.3.1 Library Audio Processing

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

Kode 2: Instalasi library audio

1.3.2 Library Image Processing

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

Kode 3: Instalasi library image

1.3.3 Library Video Processing

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

Kode 4: Instalasi library video

1.3.4 Library General Purpose

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

Kode 5: Instalasi library umum

Dokumentasikan di sini:

- Perintah instalasi yang Anda gunakan:
  - Audio: `uv pip install librosa soundfile scipy`
  - Image: `uv pip install opencv-python pillow scikit-image matplotlib`
  - Video: `uv pip install moviepy imageio-ffmpeg`
  - General: `uv pip install numpy pandas jupyter`
- Screenshot proses instalasi atau output sukses:

(a) Audio (b) Image

(c) Video (d) General

Gambar 2: Screenshots of installation steps in `uv` environment

- Daftar library yang berhasil diinstall dengan versinya:

```
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==5.2.1
defusedxml==0.7.1
exceptiongroup==1.3.0
executing==2.2.0
fastjsonschema==2.21.2
fonttools==4.59.1
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==1.5.1
json5==0.12.1
jsonpointer==3.0.0
jsonschema==4.25.1
jsonschema-specifications==2025.4.1
jupyter==1.1.1
jupyter-client==8.6.3
jupyter-console==6.6.3
jupyter-core==5.8.1
jupyter-events==0.12.0
jupyter-lsp==2.2.6
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==3.1.3
moviepy==2.2.1
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==4.3.8
pooch==1.8.2
proglog==0.1.12
prometheus-client==0.22.1
prompt-toolkit==3.0.51
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==0.27.0
scikit-image==0.25.2
scikit-learn==1.7.1
scipy==1.15.3
send2trash==1.8.3
setuptools==80.9.0
six==1.17.0
sniffio==1.3.1
soundfile==0.13.1
soupsieve==2.7
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
```

1.4 Bagian 3: Verifikasi Instalasi

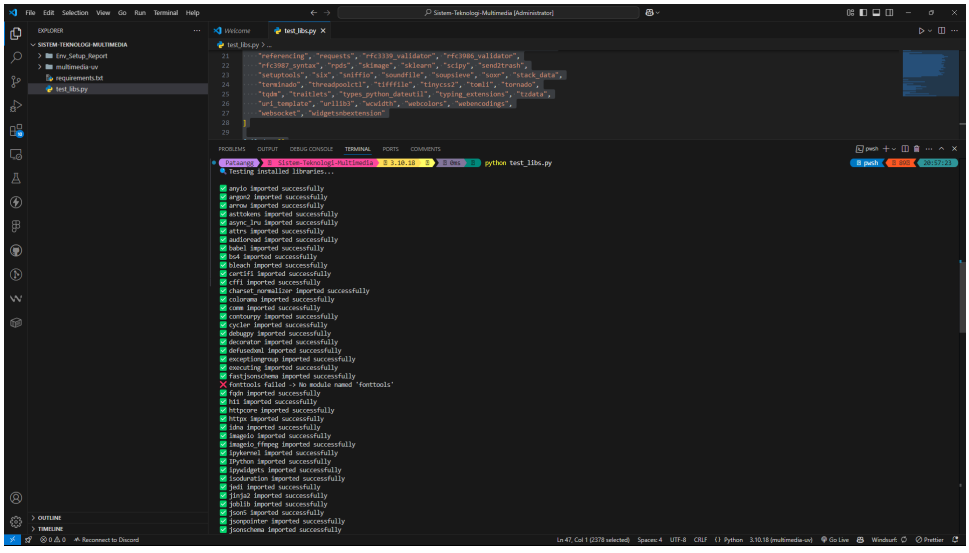
Buat file Python sederhana untuk menguji semua library yang telah diinstall:

```python
import importlib

# Daftar library dari requirements.txt
libraries = [
    "anyio", "argon2", "arrow", "asttokens", "async_lru", "attrs", "audioread",
    "babel", "bs4", "bleach", "certifi", "cffi", "charset_normalizer", "colorama",
    "comm", "contourpy", "cycler", "debugpy", "decorator", "defusedxml", "exceptiongroup",
    "executing", "fastjsonschema", "fonttools", "fqdn", "h11", "httpcore", "httpx",
    "idna", "imageio", "imageio_ffmpeg", "ipykernel", "IPython", "ipywidgets",
    "isoduration", "jedi", "jinja2", "joblib", "json5", "jsonpointer", "jsonschema",
    "jupyter", "jupyter_client", "jupyter_console", "jupyter_core", "jupyter_events",
    "jupyter_lsp", "jupyter_server", "jupyter_server_terminals", "jupyterlab",
    "jupyterlab_pygments", "jupyterlab_server", "jupyterlab_widgets", "kiwisolver",
    "lark", "lazy_loader", "librosa", "llvmlite", "markupsafe", "matplotlib",
    "matplotlib_inline", "mistune", "moviepy", "msgpack", "nbclient", "nbconvert",
    "nbformat", "nest_asyncio", "networkx", "notebook", "notebook_shim", "numba",
    "numpy", "cv2", "overrides", "packaging", "pandas", "pandocfilters", "parso",
    "PIL", "platformdirs", "pooch", "proglog", "prometheus_client", "prompt_toolkit",
    "psutil", "pure_eval", "pycparser", "pygments", "pyparsing", "dateutil",
```

```
    "dotenv", "python_json_logger", "pytz", "pywin32", "pywinpty", "yaml", "zmq",
    "referencing", "requests", "rfc3339_validator", "rfc3986_validator",
    "rfc3987_syntax", "rpds", "skimage", "sklearn", "scipy", "send2trash",
    "setuptools", "six", "sniffio", "soundfile", "soupsieve", "soxr", "stack_data",
    "terminado", "threadpoolctl", "tifffile", "tinycss2", "tomli", "tornado",
    "tqdm", "traitlets", "types_python_dateutil", "typing_extensions", "tzdata",
    "uri_template", "urllib3", "wcwidth", "webcolors", "webencodings",
    "websocket", "widgetsnbextension"
]

failed = []

print("     Testing installed libraries...\n")

for lib in libraries:
    try:
        importlib.import_module(lib)
        print(f"    {lib} imported successfully")
    except Exception as e:
        print(f"    {lib} failed -> {e}")
        failed.append(lib)

print("\n=== SUMMARY ===")
if not failed:
    print("     All libraries imported successfully!")
else:
    print(f"       Failed to import {len(failed)} libraries: {failed}")
```

Kode 6: Kode Python Sederhana untuk Cek library terinstall

Jalankan script dan dokumentasikan hasilnya:



Gambar 3: Script Python sederhana untuk cek library

1.5 Bagian 4: Simple Test dengan Sample Code

Buat dan jalankan contoh sederhana untuk setiap kategori multimedia:

1.5.1 Test Audio Processing

```
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 7: Test audio processing sederhana

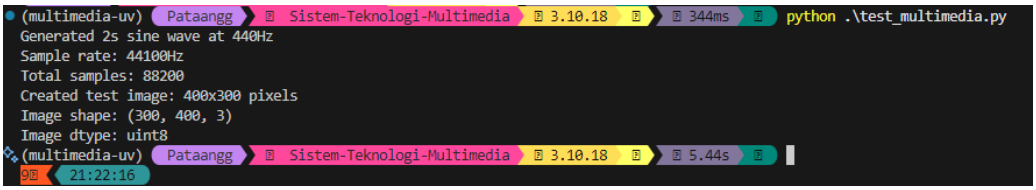
1.5.2 Test Image Processing

```
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}")
```

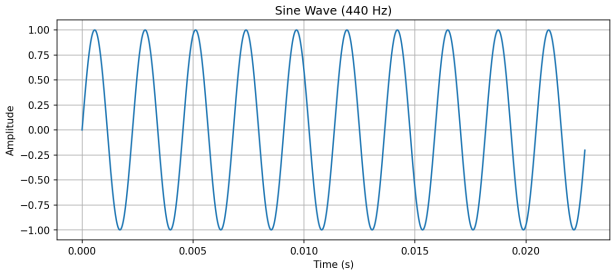
Kode 8: Test image processing sederhana

Dokumentasikan hasil eksekusi:

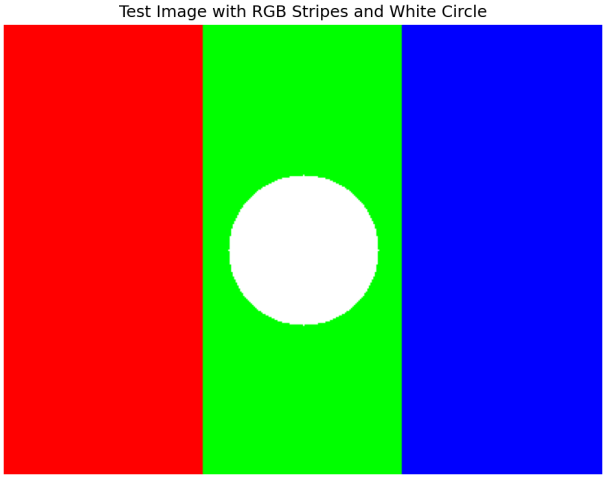
- Screenshot output dari kedua script di atas



Gambar 4: Screenshot output terminal dari kedua script



(a) Sine Wave



(b) RGB

Gambar 5: Gambar yang dihasilkan (sine_wave_test.png dan test_image.png)

2 Bagian Laporan

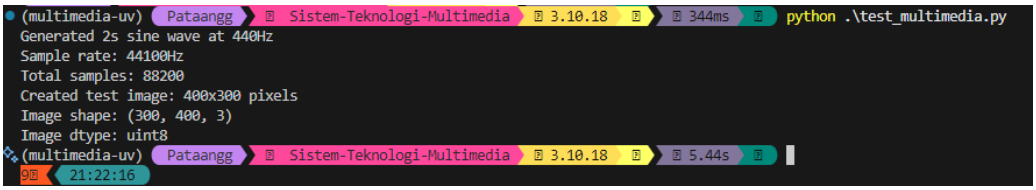
2.1 Output Verifikasi Instalasi

Copy-paste output lengkap dari script `test_multimedia.py` di sini:

```
[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]
```

Kode 9: Output verifikasi instalasi

2.2 Screenshot Hasil Test



Gambar 6: Screenshot output terminal dari kedua script

2.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

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

   *Untuk **isolasi dependensi**. Supaya gaada konflik antar proyek yang membutuhkan versi library yang berbeda, memastikan **reproducibility** (proyek dapat dijalankan di mana saja dengan hasil yang sama), dan menjaga instalasi Python global tetap bersih*

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

   `venv` *Bawaan Python, ringan, hanya mengelola paket Python*

   `conda` *Mengelola paket Python, versi Python, dan dependensi non-Python (misal: CUDA). Sangat kuat namun lebih lambat*

   `uv` *Installer dan manajer environment modern yang **sangat cepat** karena ditulis dalam Rust*

   *`uv` dipilih karena **kecepatannya** yang superior dalam menginstal dan menyelesaikan dependensi, sangat menghemat waktu pada proyek dengan banyak library*

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

   Gaada yang sulit sih semuanya bisa terinstall dengan baik

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

   - *Baca pesan error untuk mengidentifikasi paket yang konflik*
   - *Coba upgrade paket utama yang menyebabkan masalah*
   - *Tentukan versi paket yang kompatibel untuk semua dependensi secara manual*
   - *Gunakan tool manajemen seperti `poetry` buat lock versi library*
   - *Kalo semua gagal, buat ulang environment dari awal wkwkwk*

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

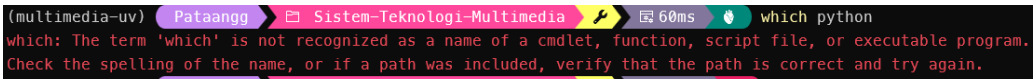
   - ***Audio Processing***
     - *`librosa`: Analisis dan ekstraksi fitur audio (spektogram, MFCC)*
     - *`soundfile`: Membaca dan menulis berbagai format file audio*
     - *`scipy`: Komputasi saintifik dan pemrosesan sinyal digital*
   - ***Image Processing***
     - *`opencv-python`: Pustaka utama untuk computer vision*

  - `Pillow`: *Manipulasi gambar dasar (crop, resize, rotasi)*
  - `scikit-image`: *Algoritma analisis citra untuk riset*
  - `matplotlib`: *Visualisasi data dan menampilkan gambar*
- ***Video Processing***
  - `moviepy`: *Editing video melalui skrip (memotong, menggabung)*
  - `imageio-ffmpeg`: *Wrapper untuk FFmpeg agar bisa membaca/menulis format video*
- ***General Purpose***
  - `numpy`: *Komputasi numerik fundamental dengan array N-dimensi*
  - `pandas`: *Analisis dan manipulasi data terstruktur (tabel)*
  - `jupyter`: *Lingkungan pemrograman interaktif (Notebook)*

2.4 Troubleshooting

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

- **Masalah 1:** *["Which" command tidak dikenali]*



Gambar 7: Which command invalid

Solusi:

- Step 1. Tanya GPT-5
- Step 2. Ternyata masalahnya karena saya pake windows dan command "Which" itu cuma untuk MacOS dan Linux
- Step 3. Akhirnya saya coba command "where python" ternyata gabisa juga. Sebab command itu bisanya dipake di CMD sedangkan saya pakenya terminal (poweshell)

3 Export Environment untuk Reproduksi

Sebagai langkah terakhir, export environment Anda agar dapat direproduksi:

3.1 Untuk venv/uv

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

Kode 10: Export pip requirements

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

```
[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==5.2.1
defusedxml==0.7.1
exceptiongroup==1.3.0
executing==2.2.0
fastjsonschema==2.21.2
fonttools==4.59.1
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==1.5.1
json5==0.12.1
jsonpointer==3.0.0
jsonschema==4.25.1
jsonschema-specifications==2025.4.1
jupyter==1.1.1
jupyter-client==8.6.3
jupyter-console==6.6.3
jupyter-core==5.8.1
jupyter-events==0.12.0
jupyter-lsp==2.2.6
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==3.1.3
moviepy==2.2.1
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==4.3.8
pooch==1.8.2
proglog==0.1.12
prometheus-client==0.22.1
prompt-toolkit==3.0.51
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==0.27.0
scikit-image==0.25.2
scikit-learn==1.7.1
scipy==1.15.3
send2trash==1.8.3
setuptools==80.9.0
six==1.17.0
sniffio==1.3.1
soundfile==0.13.1
soupsieve==2.7
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
]
```

Kode 11: Environment/Requirements file

4 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

Setup environment Python buat multimedia itu wajib hukumnya biar nggak ribet sama konflik library. Makanya pakai `uv` *atau* `venv` *itu bukan sekadar opsional, tapi memang keharusan. Dari pengalaman,* `uv` *kerasa banget cepetnya pas install paket gede kayak* `OpenCV` *dan* `SciPy`

Biar aman ke depannya:

1. Selalu mulai dari environment baru yang bersih
2. Habis install, langsung simpan `requirements.txt` pakai `uv pip freeze > requirements.txt`
3. Pahami fungsi inti tiap library biar nggak salah pilih alat

Tips buat temen-temen lain:

- Jangan install global, bikin environment dari awal
- Lebih enak langsung pakai `uv`, simpel dan cepat
- Kalau error pas install, baca pesan error dulu—biasanya jelas apa yang kurang
- Install library seperlunya aja, biar gampang kalau ada masalah

5 Referensi

- UV Official Documentation
- ChatGPT
- LaTeX Wikibook: Comprehensive Guide
- Overleaf Learn LaTeX Documentation

6 Lampiran

- Github