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

Mata Kuliah: **Sistem Teknologi Multimedia (IF25-40305)**   Tanggal: August 28, 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

```
1  # Install uv terlebih dahulu jika belum ada
2  pip install uv
3
4  # bisa dengan powershell (Windows)
5  powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
6
7  # atau terminal (MacOS dan Linux)
8  curl -LsSf https://astral.sh/uv/install.sh | sh
9
10 # Membuat environment baru
11 uv venv multimedia-uv
12
13 # Mengaktifkan environment (Linux/Mac)
14 source multimedia-uv/bin/activate
15
16 # Mengaktifkan environment (Windows)
17 multimedia-uv\Scripts\activate
18
19 # Verifikasi environment aktif (MacOS dan Linux)
20 which python
21
22 # Verifikasi environtment aktif (Windows)
23 ## Jika Menggunakan CMD
24 where python
25
26 ## Jika Menggunakan Powershell
27 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

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

Kode 2: Instalasi library audio

1.3.2 Library Image Processing

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

Kode 3: Instalasi library image

Worksheet 1: Setup Python Environment untuk Multimedia

1.3.3 Library Video Processing

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

Kode 4: Instalasi library video

1.3.4 Library General Purpose

```
1 # Untuk pip (venv/uv):
2 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:
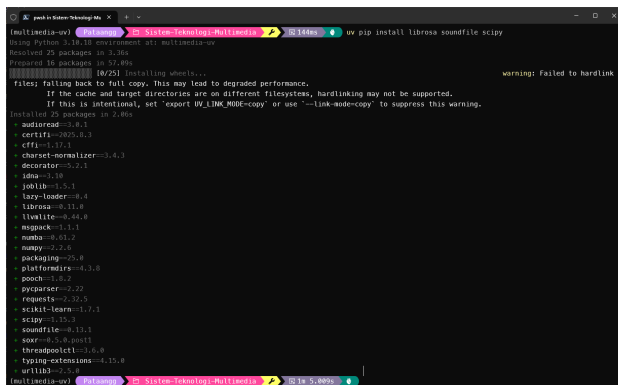
(a) Audio
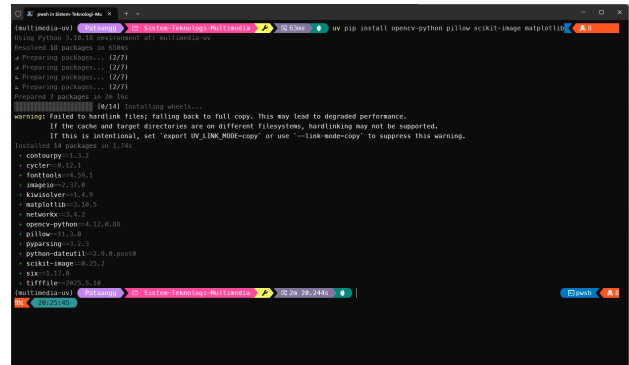
(b) Image
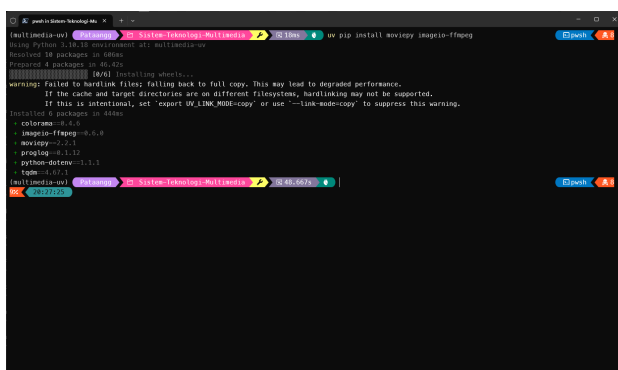
(c) Video
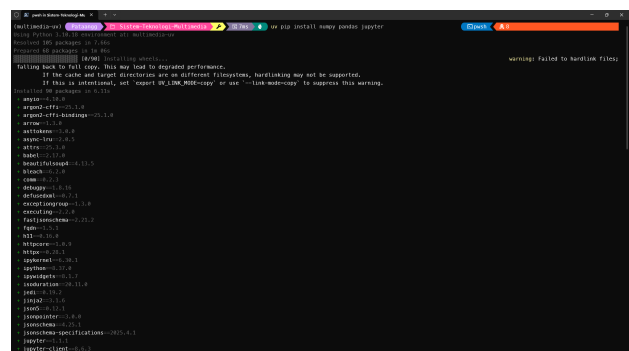
(d) General

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

- Daftar library yang berhasil diinstall dengan versinya:

Worksheet 1: Setup Python Environment untuk Multimedia

```
anyio==4.10.0                        jupyter-console==6.6.3              pygments==2.19.2
argon2-cffi==25.1.0                  jupyter-core==5.8.1                 pyparsing==3.2.3
argon2-cffi-bindings==25.1.0         jupyter-events==0.12.0             python-dateutil==2.9.0.post0
arrow==1.3.0                         jupyter-lsp==2.2.6                  python-dotenv==1.1.1
asttokens==3.0.0                     jupyter-server==2.17.0             python-json-logger==3.3.0
async-lru==2.0.5                     jupyter-server-terminals==0.5.3    pytz==2025.2
attrs==25.3.0                        jupyterlab==4.4.6                  pywin32==311
audioread==3.0.1                     jupyterlab-pygments==0.3.0         pywinpty==3.0.0
babel==2.17.0                        jupyterlab-server==2.27.3          pyyaml==6.0.2
beautifulsoup4==4.13.5               jupyterlab-widgets==3.0.15         pyzmq==27.0.2
bleach==6.2.0                        kiwisolver==1.4.9                 referencing==0.36.2
certifi==2025.8.3                    lark==1.2.2                       requests==2.32.5
cffi==1.17.1                         lazy-loader==0.4                  rfc3339-validator==0.1.4
charset-normalizer==3.4.3            librosa==0.11.0                   rfc3986-validator==0.1.1
colorama==0.4.6                      llvmlite==0.44.0                  rfc3987-syntax==1.1.0
comm==0.2.3                          markupsafe==3.0.2                 rpds-py==0.27.0
contourpy==1.3.2                     matplotlib==3.10.5                scikit-image==0.25.2
cycler==0.12.1                       matplotlib-inline==0.1.7          scikit-learn==1.7.1
debugpy==1.8.16                      mistune==3.1.3                    scipy==1.15.3
decorator==5.2.1                     moviepy==2.2.1                    send2trash==1.8.3
defusedxml==0.7.1                    msgpack==1.1.1                    setuptools==80.9.0
exceptiongroup==1.3.0                nbclient==0.10.2                  six==1.17.0
executing==2.2.0                     nbconvert==7.16.6                 sniffio==1.3.1
fastjsonschema==2.21.2               nbformat==5.10.4                  soundfile==0.13.1
fonttools==4.59.1                    nest-asyncio==1.6.0               soupsieve==2.7
fqdn==1.5.1                          networkx==3.4.2                   soxr==0.5.0.post1
h11==0.16.0                          notebook==7.4.5                   stack-data==0.6.3
httpcore==1.0.9                      notebook-shim==0.2.4              terminado==0.18.1
httpx==0.28.1                        numba==0.61.2                     threadpoolctl==3.6.0
idna==3.10                           numpy==2.2.6                      tifffile==2025.5.10
imageio==2.37.0                      opencv-python==4.12.0.88          tinycss2==1.4.0
imageio-ffmpeg==0.6.0                overrides==7.7.0                  tomli==2.2.1
ipykernel==6.30.1                    packaging==25.0                   tornado==6.5.2
ipython==8.37.0                      pandas==2.3.2                     tqdm==4.67.1
ipywidgets==8.1.7                    pandocfilters==1.5.1              traitlets==5.14.3
isoduration==20.11.0                 parso==0.8.5                      types-python-dateutil==2.9.0.20250822
jedi==0.19.2                         pillow==11.3.0                    typing-extensions==4.15.0
jinja2==3.1.6                        platformdirs==4.3.8               tzdata==2025.2
joblib==1.5.1                        pooch==1.8.2                      uri-template==1.3.0
json5==0.12.1                        proglog==0.1.12                   urllib3==2.5.0
jsonpointer==3.0.0                   prometheus-client==0.22.1         wcwidth==0.2.13
jsonschema==4.25.1                   prompt-toolkit==3.0.51            webcolors==24.11.1
jsonschema-specifications==2025.4.1  psutil==7.0.0                     webencodings==0.5.1
jupyter==1.1.1                       pure-eval==0.2.3                  websocket-client==1.8.0
jupyter-client==8.6.3                pycparser==2.22                   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",
```

Worksheet 1: Setup Python Environment untuk Multimedia

```
20      "dotenv", "python_json_logger", "pytz", "pywin32", "pywinpty", "yaml", "zmq",
21      "referencing", "requests", "rfc3339_validator", "rfc3986_validator",
22      "rfc3987_syntax", "rpds", "skimage", "sklearn", "scipy", "send2trash",
23      "setuptools", "six", "sniffio", "soundfile", "soupsieve", "soxr", "stack_data",
24      "terminado", "threadpoolctl", "tifffile", "tinycss2", "tomli", "tornado",
25      "tqdm", "traitlets", "types_python_dateutil", "typing_extensions", "tzdata",
26      "uri_template", "urllib3", "wcwidth", "webcolors", "webencodings",
27      "websocket", "widgetsnbextension"
28  ]
29
30  failed = []
31
32  print("Testing installed libraries...\n")
33  print("Testing installed libraries...\n")
34
35  for lib in libraries:
36      try:
37          importlib.import_module(lib)
38          print(f"[OK] {lib} imported successfully")
39      except Exception as e:
40          print(f"[FAIL] {lib} failed -> {e}")
41          failed.append(lib)
42
43  print("\n=== SUMMARY ===")
44  if not failed:
45      print("All libraries imported successfully!")
46  else:
47      print(f"Failed to import {len(failed)} libraries: {failed}")
```

Kode 6: Kode Python Sederhana untuk Cek library terinstall

Jalankan script dan dokumentasikan hasilnya:
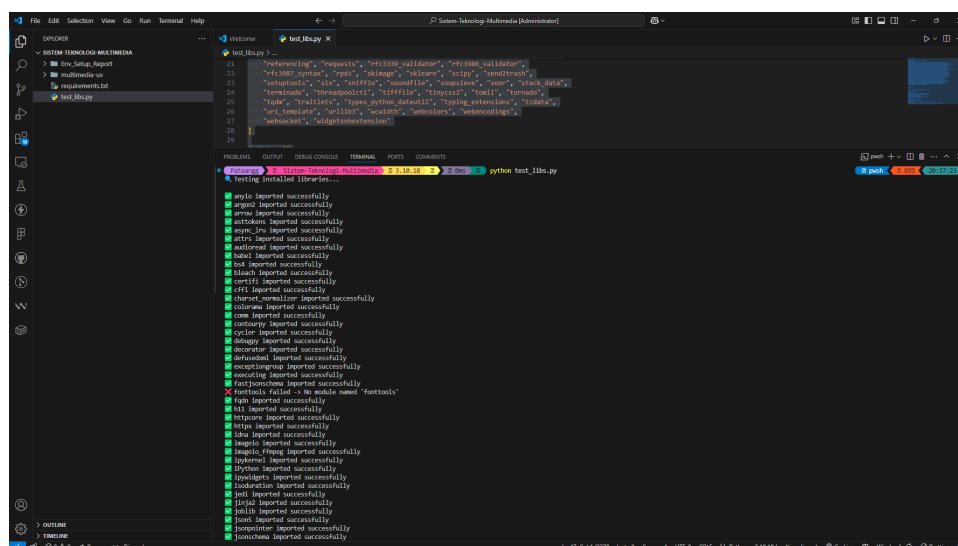
Gambar 3: Script Python sederhana untuk cek library

1.5   Bagian 4: Simple Test dengan Sample Code

Buat dan jalankan contoh sederhana untuk setiap kategori multimedia:

1.5.1   Test Audio Processing

```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 7: Test audio processing sederhana

1.5.2 Test Image Processing

```python
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}")
```

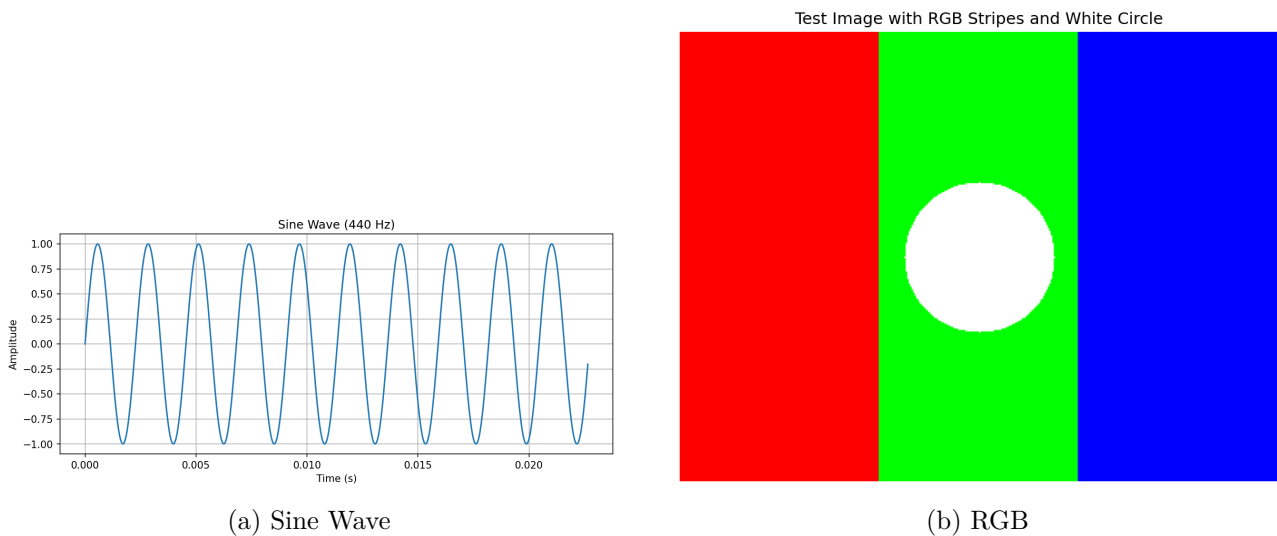Kode 8: Test image processing sederhana

Worksheet 1: Setup Python Environment untuk Multimedia

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

2.3   Screenshot Hasil Test

Gambar 6: Screenshot output terminal dari kedua script

2.4   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*

**Worksheet 1: Setup Python Environment untuk Multimedia**

- *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)*

2.5 Troubleshooting

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

- **Masalah 1:** *["Which" command tidak dikenali]*
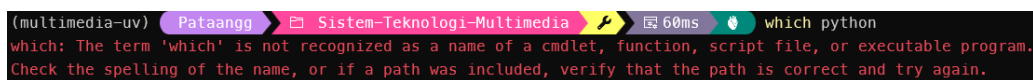
Gambar 7: Which command invalid

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
```

Worksheet 1: Setup Python Environment untuk Multimedia

```
10  beautifulsoup4==4.13.5
11  bleach==6.2.0
12  certifi==2025.8.3
13  cffi==1.17.1
14  charset-normalizer==3.4.3
15  colorama==0.4.6
16  comm==0.2.3
17  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
24  fastjsonschema==2.21.2
25  fonttools==4.59.1
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  imageio==2.37.0
32  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==1.5.1
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-client==8.6.3
46  jupyter-console==6.6.3
47  jupyter-core==5.8.1
48  jupyter-events==0.12.0
49  jupyter-lsp==2.2.6
50  jupyter-server==2.17.0
51  jupyter-server-terminals==0.5.3
52  jupyterlab==4.4.6
53  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==3.1.3
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
```

Worksheet 1: Setup Python Environment untuk Multimedia

```
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==4.3.8
84  pooch==1.8.2
85  proglog==0.1.12
86  prometheus-client==0.22.1
87  prompt-toolkit==3.0.51
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
101 referencing==0.36.2
102 requests==2.32.5
103 rfc3339-validator==0.1.4
104 rfc3986-validator==0.1.1
105 rfc3987-syntax==1.1.0
106 rpds-py==0.27.0
107 scikit-image==0.25.2
108 scikit-learn==1.7.1
109 scipy==1.15.3
110 send2trash==1.8.3
111 setuptools==80.9.0
112 six==1.17.0
113 sniffio==1.3.1
114 soundfile==0.13.1
115 soupsieve==2.7
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 tinycss2==1.4.0
122 tomli==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
```

Worksheet 1: Setup Python Environment untuk Multimedia

```
134 websocket-client==1.8.0
135 widgetsnbextension==4.0.14
136 ]
```

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](#)