



---

Nama: **Elma Nurul Fatika (122140069)** 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  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

## 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  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  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: [**Opsi 3, menggunakan uv**]
- Screenshot atau copy-paste output dari perintah verifikasi environment

```

PS C:\Multimedia\Tugas1> pip install uv
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: uv in c:\users\elma\appdata\roaming\python\python313\site-packages (0.8.13)
PS C:\Multimedia\Tugas1> uv venv multimedia-uv
Using CPython 3.10.18
Creating virtual environment at: multimedia-uv
Activate with: multimedia-uv\Scripts\activate
PS C:\Multimedia\Tugas1> multimedia-uv\Scripts\activate
(multimedia-uv) PS C:\Multimedia\Tugas1> which python
which : The term 'which' is not recognized as the name of a cmdlet, function,
script file, or operable program. Check the spelling of the name, or if a path
was included, verify that the path is correct and try again.
At line:1 char:1
+ which python
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (which:String) [], CommandNotFou
ndException
+ FullyQualifiedErrorId : CommandNotFoundException

(multimedia-uv) PS C:\Multimedia\Tugas1> python --version
Python 3.10.18
(multimedia-uv) PS C:\Multimedia\Tugas1> where python

```

saya menggunakan "which python" terjadi eror karena perintah tersebut digunakan untuk MacOS, dan untuk windows menggunakan "where python"

### 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 Instalasi library audio:
2 (multimedia-uv) PS C:\Multimedia\Tugas1> uv pip install librosa soundfile scipy
3 Using Python 3.10.18 environment at: multimedia-uv
4 Resolved 25 packages in 4.51s
5 Prepared 13 packages in 8m 22s
6 Installed 13 packages in 539ms
7 + audioread==3.0.1
8 + joblib==1.5.2
9 + lazy-loader==0.4
10 + librosa==0.11.0
11 + llvmlite==0.44.0
12 + msgpack==1.1.1
13 + numba==0.61.2
14 + pooch==1.8.2
15 + scikit-learn==1.7.1
16 + scipy==1.15.3
17 + soundfile==0.13.1
18 + soxr==0.5.0.post1
19 + threadpoolctl==3.6.0
20
21 Instalasi library image:
22 (multimedia-uv) PS C:\Multimedia\Tugas1> uv pip install opencv-python pillow scikit-image
23 matplotlib
24 Using Python 3.10.18 environment at: multimedia-uv
25 Resolved 18 packages in 14.28s
26 Resolved 18 packages in 14.28s
27 Prepared 5 packages in 7m 22s
28 Installed 12 packages in 452ms
29 + contourpy==1.3.2
30 + cycler==0.12.1
31 + fonttools==4.59.2
32 + imageio==2.37.0
33 + kiwisolver==1.4.9
34 + matplotlib==3.10.5
35 + networkx==3.4.2
36 + opencv-python==4.12.0.88
37 + pillow==11.3.0
38 + pyparsing==3.2.3
39 + scikit-image==0.25.2
40 + tifffile==2025.5.10
41
42 Instalasi library video:
43 (multimedia-uv) PS C:\Multimedia\Tugas1> uv pip install moviepy
44 Using Python 3.10.18 environment at: multimedia-uv

```

```
45 Resolved 10 packages in 781ms
46 Resolved 10 packages in 781ms
47 Prepared 4 packages in 2m 15s
48 Installed 5 packages in 926ms
49 + imageio-ffmpeg==0.6.0
50 + moviepy==2.2.1
51 + proglog==0.1.12
52 + python-dotenv==1.1.1
53 + tqdm==4.67.1
54
55 Instalasi library umum:
56 (multimedia-uv) PS C:\Multimedia\Tugas1> uv pip install numpy pandas jupyter
57 Using Python 3.10.18 environment at: multimedia-uv
58 Resolved 105 packages in 2.54s
59 Prepared 9 packages in 2m 06s
60 Installed 75 packages in 2.03s
61 + anyio==4.10.0
62 + argon2-cffi==25.1.0
63 + argon2-cffi-bindings==25.1.0
64 + arrow==1.3.0
65 + async-lru==2.0.5
66 + attrs==25.3.0
67 + babel==2.17.0
68 + beautifulsoup4==4.13.5
69 + bleach==6.2.0
70 + certifi==2025.8.3
71 + cffi==1.17.1
72 + charset-normalizer==3.4.3
73 + defusedxml==0.7.1
74 + fastjsonschema==2.21.2
75 + fqdn==1.5.1
76 + h11==0.16.0
77 + httpcore==1.0.9
78 + httpx==0.28.1
79 + idna==3.10
80 + ipywidgets==8.1.7
81 + isoduration==20.11.0
82 + jinja2==3.1.6
83 + json5==0.12.1
84 + jsonpointer==3.0.0
85 + jsonschema==4.25.1
86 + jsonschema-specifications==2025.4.1
87 + jupyter==1.1.1
88 + jupyter-console==6.6.3
89 + jupyter-events==0.12.0
90 + jupyter-lsp==2.3.0
91 + jupyter-server==2.17.0
92 + jupyter-server-terminals==0.5.3
93 + jupyterlab==4.4.6
94 + jupyterlab-pygments==0.3.0
95 + jupyterlab-server==2.27.3
96 + jupyterlab-widgets==3.0.15
97 + lark==1.2.2
98 + markupsafe==3.0.2
99 + mistune==3.1.3
100 + nbclient==0.10.2
101 + nbconvert==7.16.6
102 + nbformat==5.10.4
103 + notebook==7.4.5
104 + notebook-shim==0.2.4
105 + numpy==2.2.6
106 + overrides==7.7.0
```

```
107 + pandas==2.3.2
108 + pandocfilters==1.5.1
109 + prometheus-client==0.22.1
110 + pycparser==2.22
111 + python-json-logger==3.3.0
112 + pytz==2025.2
113 + pywinpty==3.0.0
114 + pyyaml==6.0.2
115 + referencing==0.36.2
116 + requests==2.32.5
117 + rfc3339-validator==0.1.4
118 + rfc3986-validator==0.1.1
119 + rfc3987-syntax==1.1.0
120 + rpds-py==0.27.1
121 + send2trash==1.8.3
122 + setuptools==80.9.0
123 + pytz==2025.2
124 + pywinpty==3.0.0
125 + pyyaml==6.0.2
126 + referencing==0.36.2
127 + requests==2.32.5
128 + rfc3339-validator==0.1.4
129 + rfc3986-validator==0.1.1
130 + rfc3987-syntax==1.1.0
131 + rpds-py==0.27.1
132 + send2trash==1.8.3
133 + setuptools==80.9.0
134 + pywinpty==3.0.0
135 + pyyaml==6.0.2
136 + referencing==0.36.2
137 + requests==2.32.5
138 + rfc3339-validator==0.1.4
139 + rfc3986-validator==0.1.1
140 + rfc3987-syntax==1.1.0
141 + rpds-py==0.27.1
142 + send2trash==1.8.3
143 + setuptools==80.9.0
144 + referencing==0.36.2
145 + requests==2.32.5
146 + rfc3339-validator==0.1.4
147 + rfc3986-validator==0.1.1
148 + rfc3987-syntax==1.1.0
149 + rpds-py==0.27.1
150 + send2trash==1.8.3
151 + setuptools==80.9.0
152 + rfc3339-validator==0.1.4
153 + rfc3986-validator==0.1.1
154 + rfc3987-syntax==1.1.0
155 + rpds-py==0.27.1
156 + send2trash==1.8.3
157 + setuptools==80.9.0
158 + sniffio==1.3.1
159 + soupsieve==2.8
160 + terminado==0.18.1
161 + rfc3987-syntax==1.1.0
162 + rpds-py==0.27.1
163 + send2trash==1.8.3
164 + setuptools==80.9.0
165 + sniffio==1.3.1
166 + soupsieve==2.8
167 + terminado==0.18.1
168 + send2trash==1.8.3
```

```

169 + setuptools==80.9.0
170 + sniffio==1.3.1
171 + soupsieve==2.8
172 + terminado==0.18.1
173 + tinycss2==1.4.0
174 + sniffio==1.3.1
175 + soupsieve==2.8
176 + terminado==0.18.1
177 + tinycss2==1.4.0
178 + tinycss2==1.4.0
179 + tomli==2.2.1
180 + tomli==2.2.1
181 + types-python-dateutil==2.9.0.20250822
182 + tzdata==2025.2
183 + types-python-dateutil==2.9.0.20250822
184 + tzdata==2025.2
185 + uri-template==1.3.0
186 + types-python-dateutil==2.9.0.20250822
187 + tzdata==2025.2
188 + types-python-dateutil==2.9.0.20250822
189 + tzdata==2025.2
190 + uri-template==1.3.0
191 + urllib3==2.5.0
192 + webcolors==24.11.1
193 + webencodings==0.5.1
194 + websocket-client==1.8.0
195 + widgetsnextension==4.0.14
196

```

Kode 8: 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:**

**File dan dokumentasi hasil eksekusi:**

```

1 import importlib
2
3 libraries = [
4     ("librosa", "Audio"),
5     ("soundfile", "Audio"),
6     ("scipy", "Audio"),
7     ("cv2", "Image"),
8     ("PIL", "Image"),
9     ("skimage", "Image"),
10    ("matplotlib", "Image"),
11    ("moviepy", "Video"),
12    ("numpy", "General"),
13    ("pandas", "General"),
14    ("jupyter", "General"),
15 ]
16
17 print("Cek instalasi library multimedia:\n")
18
19 for lib, category in libraries:
20     try:
21         module = importlib.import_module(lib)
22         version = getattr(module, "__version__", "versi tidak diketahui")
23         print(f"[SUKSES] {lib} ({category}) terinstall, versi: {version}")
24     except Exception as e:

```

```
25 print(f"[GAGAL] {lib} ({category}) TIDAK terinstall. ({e})")
```

Kode 9: verifikasi instalasi library multimedia

```
PS C:\Multimedia\Tugas1> & "C:/Program Files/Python313/python.exe" c:/Multimedia/Tugas1/test_multimedia.py
Cek instalasi library multimedia:

[SUKSES] librosa (Audio) terinstall, versi: 0.11.0
[SUKSES] soundfile (Audio) terinstall, versi: 0.13.1
[SUKSES] scipy (Audio) terinstall, versi: 1.16.1
[SUKSES] cv2 (Image) terinstall, versi: 4.12.0
[SUKSES] PIL (Image) terinstall, versi: 11.3.0
[SUKSES] skimage (Image) terinstall, versi: 0.25.2
[SUKSES] matplotlib (Image) terinstall, versi: 3.10.5
[SUKSES] moviepy (Video) terinstall, versi: 2.1.2
[SUKSES] numpy (General) terinstall, versi: 2.2.6
[SUKSES] pandas (General) terinstall, versi: 2.3.2
[SUKSES] jupyter (General) terinstall, versi: versi tidak diketahui
```

### 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 10: 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 11: 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 [Cek instalasi library multimedia:
2
3 [SUKSES] librosa (Audio) terinstall, versi: 0.11.0
4 [SUKSES] soundfile (Audio) terinstall, versi: 0.13.1
5 [SUKSES] scipy (Audio) terinstall, versi: 1.16.1
6 [SUKSES] cv2 (Image) terinstall, versi: 4.12.0
7 [SUKSES] librosa (Audio) terinstall, versi: 0.11.0
8 [SUKSES] soundfile (Audio) terinstall, versi: 0.13.1
9 [SUKSES] scipy (Audio) terinstall, versi: 1.16.1
10 [SUKSES] cv2 (Image) terinstall, versi: 4.12.0
11 [SUKSES] scipy (Audio) terinstall, versi: 1.16.1
12 [SUKSES] cv2 (Image) terinstall, versi: 4.12.0
13 [SUKSES] PIL (Image) terinstall, versi: 11.3.0
14 [SUKSES] skimage (Image) terinstall, versi: 0.25.2
15 [SUKSES] skimage (Image) terinstall, versi: 0.25.2
16 [SUKSES] matplotlib (Image) terinstall, versi: 3.10.5
17 [SUKSES] moviepy (Video) terinstall, versi: 2.1.2
18 [SUKSES] moviepy (Video) terinstall, versi: 2.1.2
19 [SUKSES] numpy (General) terinstall, versi: 2.2.6
20 [SUKSES] pandas (General) terinstall, versi: 2.3.2
21 [SUKSES] jupyter (General) terinstall, versi: versi tidak diketahui]

```

Kode 12: Output verifikasi instalasi

## 4.2 Screenshot Hasil Test

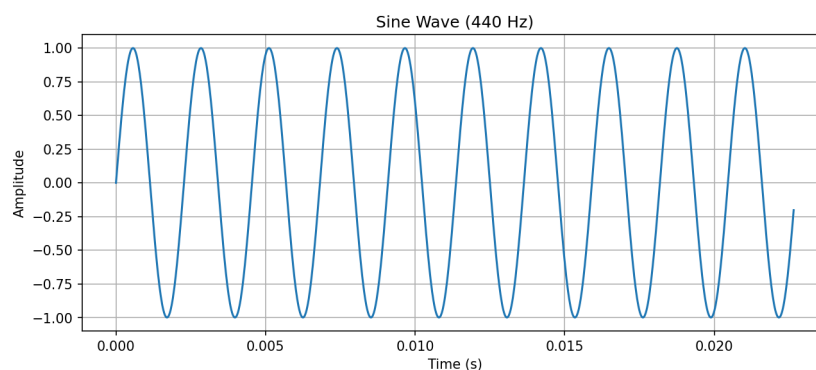
Sisipkan screenshot atau gambar hasil dari:

- Terminal/command prompt yang menunjukkan environment aktif

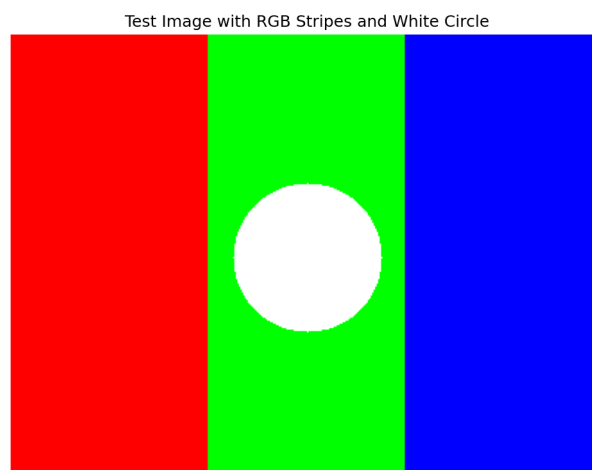
```
PS C:\Multimedia\Tugas1> pip install uv
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: uv in c:\users\elma\appdata\roaming\python\python313\site-packages (0.8.13)
PS C:\Multimedia\Tugas1> uv venv multimedia-uv
Using CPython 3.10.18
Creating virtual environment at: multimedia-uv
Activate with: multimedia-uv\Scripts\activate
PS C:\Multimedia\Tugas1> multimedia-uv\Scripts\activate
(multimedia-uv) PS C:\Multimedia\Tugas1> which python
which : The term 'which' is not recognized as the name of a cmdlet, function,
script file, or operable program. Check the spelling of the name, or if a path
was included, verify that the path is correct and try again.
At line:1 char:1
+ ~~~~~
+ which python
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (which:String) [], CommandNotFou
ndException
+ FullyQualifiedErrorId : CommandNotFoundException

(multimedia-uv) PS C:\Multimedia\Tugas1> python --version
Python 3.10.18
(multimedia-uv) PS C:\Multimedia\Tugas1> where python
```

- Output dari script test audio (sine wave plot)



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



### 4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

1. Mengapa penting menggunakan environment terpisah untuk project multimedia?

*Menggunakan environment terpisah itu supaya tidak terjadi bentrok antar library yang dipakai. jadi semisal membuat project yang lain lagi atau yang berbeda itu tidak berpengaruh pada prproject yang sudah pernah dibuat. kaarena tiap project itu beda beda library yang dipakai.*

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

*Kalau conda itu lebih lengkap dan cukup direkomendasikan untuk pemula, karena conda itu sudah include package manager dan environment manager. kalau venv itu built in python, tidak perlu install tambahan. kalau uv itu modern dan cepat tapi perlu install dulu. sayamemilih uv karena yang pertama disarankan oleh pak Martin karena lebih ringan dan cepat.*

3. Library mana yang paling sulit diinstall dan mengapa?

*Kalau menurut saya sepertinya conda, karena dilihat dari proses instalasinya yang cukup panjang dan lama. mungkin karena di conda banyak package yang harus di install.*

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

*Saya bertanya kepada chat gpt dan copilot, yang saya lakukan adalah menginstall ulang library yang bermasalah.*

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

*Librosa untuk audio processing, soundfile untuk membaca dan menulis file audio, scipy untuk manipulasi sinyal audio, cv2 untuk image processing, PIL untuk manipulasi gambar,skimage untuk analisis gambar, matplotlib untuk visualisasi data, moviepy untuk edit video, numpy untuk komputasi numerik, pandas untuk manipulasi data, jupyter untuk notebook interaktif.*

### 4.4 Troubleshooting

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

- **Masalah 1:** Ketika saya menginsatll library audio, image, video, dan umum ada beberapa library yang terinstall tapi dia berada di folder yang berbeda, dia masuk di environment laptopnya bukan di environment uv.

**Solusi:** Saya bertanya kepada chat gpt terkait warning yang ada, namun kurang membantu, kemudian saya coba untuk bertanya kepada dinda, ternyata ketika menjalankan perintah "pip install ...." itu harus diawali dengan "uv" supaya dia masuk ke environment uv, jadi baru teratasi masalahnya.

- **Masalah 2:** Ketika saya akan melakukan verifikasi environtment aktif dan mwnngikuti langkah yang tertera pada dokumen ternyata tidak bisa jika menggunakan "which python".

**Solusi:** Saya bertanya ke chat gpt dan mendapatkan informasi bahwa "which python" itu digunakan untuk MacOS, sedangkan untuk windows menggunakan "where python", dan akhirnya saya coba dan berhasil.

- **Masalah 3:** Saya belum mahir dan familiar dengan latex, jadi dalam pembuatan dokumennya saya mengalami kesulitan.

**Solusi:** Saya hanya menggunakan code code yang sudah ada dan mengcopy paste, dan juga masih banyak yang kurang rapih menurut saya.

## 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 13: Export conda environment

### 5.2 Untuk venv/uv

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

Kode 14: Export pip requirements

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

```
1 [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
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==4.59.2
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 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==1.5.2
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-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==2.3.0
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
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
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==0.27.1
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==2.8
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
134 websocket-client==1.8.0
135 widgetsnbextension==4.0.14
136 ]
```

Kode 15: 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

*Kesimpulan yang saya dapatkan adalah ketika melakukan setup python environment untuk multimedia bisa dibilang tidak mudah dan tidak sulit, mudahnya karena sudah diberikan tutorial oleh pak Martin dan sulitnya ketika mengalami trouble shooting yang tidak bisa di atasi oleh AI dan diri sendiri. Untuk persiapan selanjutnya saya akan lebih teliti dan tidak panik jika mengalami eror. Dan saran untuk teman-teman yang lain jika ingin melakukan setup serupa kalau bisa mengerjakannya bersama teman, jadi bisa saling membantu dan sama-sama belajar.*

## 7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

## References

Chat GPT : <https://chatgpt.com/share/68b0606a-78f8-8005-9ad0-2e6ed9927302>