



Nama: **Muhammad Yusuf (122140193)**

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 L^AT_EX

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 L^AT_EX 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: **[Menggunakan uv]**
- Screenshot atau copy-paste output dari perintah verifikasi environment

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\muham\OneDrive\Desktop\belajar uv> uv venv multimedia-uv
Using CPython 3.13.5 interpreter at: C:\Python313\python.exe
Creating virtual environment at: multimedia-uv
Activate with: multimedia-uv\Scripts\activate
PS C:\Users\muham\OneDrive\Desktop\belajar uv> multimedia-uv\Scripts\activate
(multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> which python
which : The term 'which' is not recognized as the name of a cmdlet, function, script file, or operable program. Che
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) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

(multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> where python
(multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> _

```

Gambar 1: Output verifikasi environment menggunakan uv

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

```
(multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> uv pip install numpy pandas jupyter
Using Python 3.13.5 environment at: multimedia-uv
Resolved 103 packages in 986ms
Prepared 99 packages in 25.16s
Installed 99 packages in 2.29s
+ anyio==4.10.0
+ argon2-cffi==25.1.0
+ argon2-cffi-bindings==25.1.0
+ arrow==1.5.1
+ asttokens==3.0.0
+ attrs==24.0.0
+ attrs==24.0.0
+ babel==2.17.0
+ beautifulsoup4==4.13.5
+ bleach==5.1.0
+ certifi==2025.8.3
+ cffi==1.17.1
+ charset-normalizer==3.4.3
+ colorama==4.6
+ comm==0.2.3
+ debugpy==1.8.16
+ decorator==5.2.1
+ defusedxml==0.7.1
+ executing==2.2.0
+ fastjsonschema==2.21.2
+ ffdm==1.5.1
+ h11==0.16.0
+ httpcore==1.0.9
+ https://0.28.1
+ idna==3.10
+ ipykernel==6.30.1
```

(a) Instalasi Library Audio (librosa, soundfile, scipy)

```
(multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> uv pip install opencv-python pillow scikit-image matplotlib
Using Python 3.13.5 environment at: multimedia-uv
Resolved 18 packages in 561ms
Prepared 18 packages in 13.34s
Installed 18 packages in 765ms
+ contourpy==1.3.3
+ cython==0.32.3
+ fonttools==4.59.2
+ imageio==2.37
+ pillow==9.4.0
+ pilLOW==9.4.0
+ lazy-loader==0.4
+ matplotlib==3.10.5
+ netwrx==3.5
+ numba==0.58.0
+ opencv-python==4.12.0.88
+ packaging==25.0
+ pillow==9.4.0
+ pydense-datatil==2.9.0.post0
+ scikit-image==0.25.2
+ scipy==1.10.1
+ six==1.16.0
+ tifffile==2025.8.28
(multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> _
```

(b) Instalasi Library Image (opencv, pillow, scikit-image, matplotlib)

```
(multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> uv pip install moviepy
Using Python 3.13.5 environment at: multimedia-uv
Resolved 10 packages in 349ms
Prepared 5 packages in 2.78s
Installed 5 packages in 43ms
+ imageio-ffmpeg==0.6.0
+ moviepy==2.2.1
+ prolog==0.1.12
+ python-dotenv==1.1.1
+ tqdm==4.67.1
(multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> _
```

(c) Instalasi Library Video (ffmpeg, moviepy)

```
(multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> uv pip install numpy pandas jupyter
Using Python 3.13.5 environment at: multimedia-uv
Audited 3 packages in 1.23s
(multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> _
```

(d) Instalasi Library Umum (numpy, pandas, jupyter)

Gambar 2: Rekaman langkah instalasi untuk seluruh kategori library multimedia.

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

```

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 8: Test audio processing sederhana

3.5.2 Test Image Processing

```

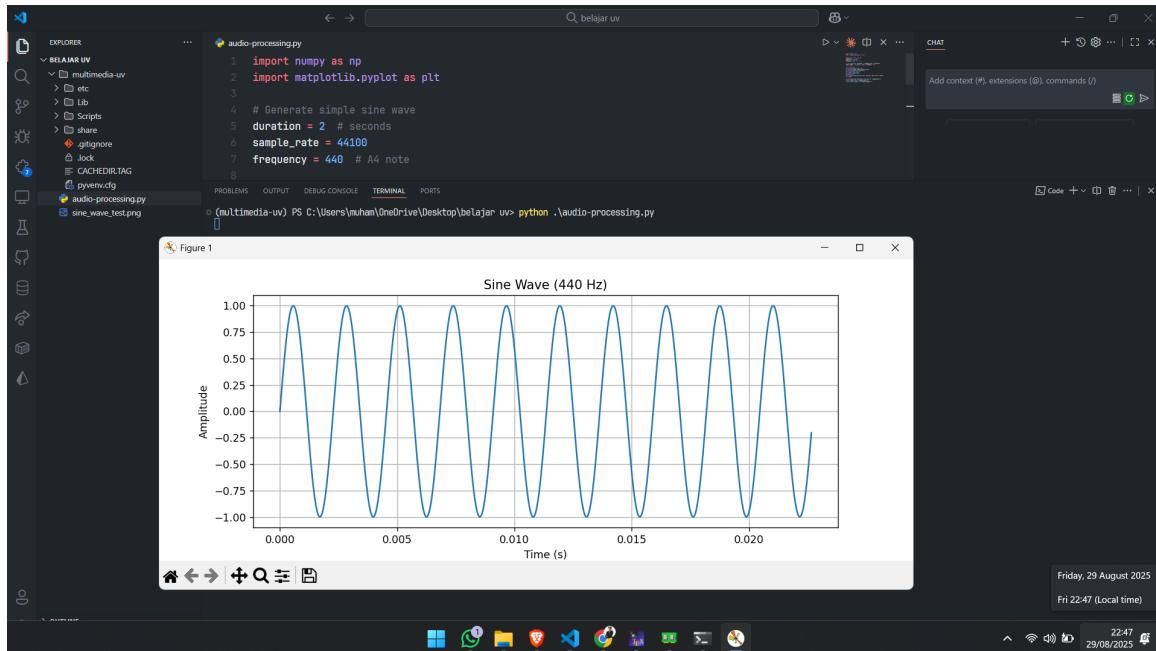
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from PIL import Image
4
5 # Create a simple test image
6 width, height = 400, 300
7 image = np.zeros((height, width, 3), dtype=np.uint8)
8
9 # Add some patterns
10 image[:, :width//3, 0] = 255 # Red section
11 image[:, width//3:2*width//3, 1] = 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}")

```

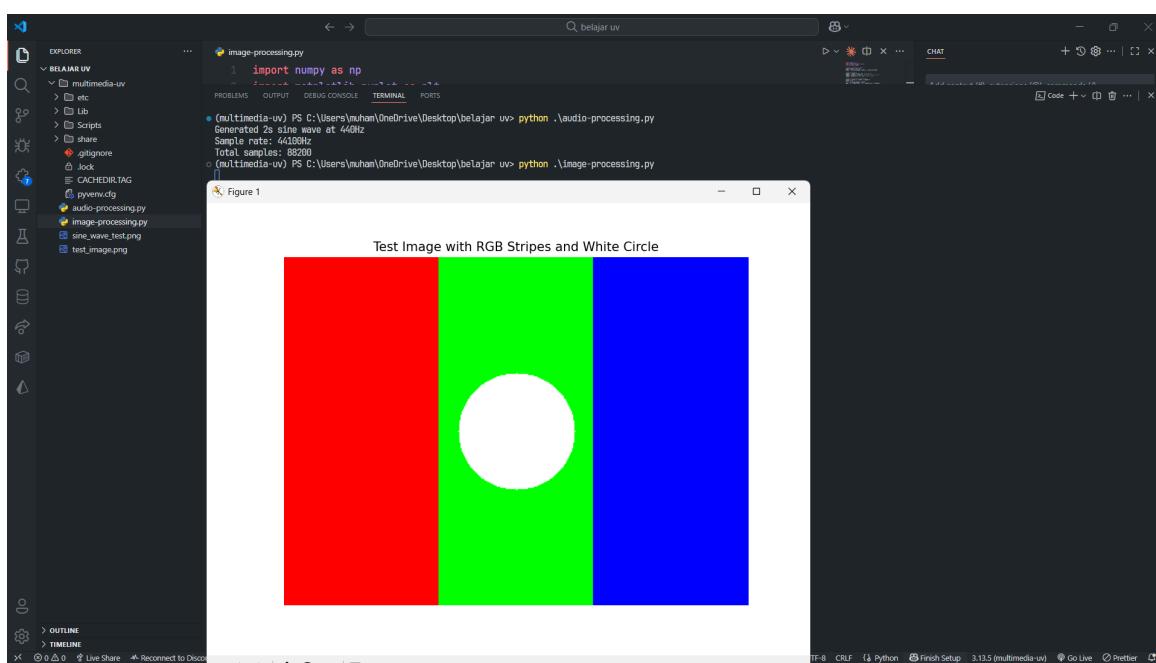
Kode 9: Test image processing sederhana

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



Gambar 3: Output code audio-processing



Gambar 4: Output code image-processing

4 Bagian Laporan

4.1 Output Verifikasi Instalasi

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

```
1 #hasil setup environment
2 PS C:\Users\muham\OneDrive\Desktop\belajar uv> uv venv multimedia-uv
3 Using CPython 3.13.5 interpreter at: C:\Python313\python.exe
4 Creating virtual environment at: multimedia-uv
5 Activate with: multimedia-uv\Scripts\activate
6 PS C:\Users\muham\OneDrive\Desktop\belajar uv> multimedia-uv\Scripts\activate
7 (multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> which python
8 which : The term 'which' is not recognized as the name of a cmdlet, function, script file, or
9 operable program. Check the
10 spelling of the name, or if a path was included, verify that the path is correct and try again.
11 At line:1 char:1
12 + which python
13 + ~~~~~
14 + CategoryInfo          : ObjectNotFound: (which:String) [], CommandNotFoundException
15 + FullyQualifiedErrorId : CommandNotFoundException
16
17 (multimedia-uv) PS C:\Users\muham\OneDrive\Desktop\belajar uv> where python
18
19 # Ada eror terminal, yang di tutorial untuk environment mac, where untuk windows
20
21 #audio-processing.py
22 import numpy as np
23 import matplotlib.pyplot as plt
24
25 # Generate simple sine wave
26 duration = 2 # seconds
27 sample_rate = 44100
28 frequency = 440 # A4 note
29
30 t = np.linspace(0, duration, int(sample_rate * duration))
31 audio_signal = np.sin(2 * np.pi * frequency * t)
32
33 # Plot waveform
34 plt.figure(figsize=(10, 4))
35 plt.plot(t[:1000], audio_signal[:1000])
36 plt.title('Sine Wave (440 Hz)')
37 plt.xlabel('Time (s)')
38 plt.ylabel('Amplitude')
39 plt.grid(True)
40 plt.savefig('sine_wave_test.png', dpi=150, bbox_inches='tight')
41 plt.show()
42
43 print(f"Generated {duration}s sine wave at {frequency}Hz")
44 print(f"Sample rate: {sample_rate}Hz")
45 print(f"Total samples: {len(audio_signal)}")
46
47 #image-processing.py
48 import numpy as np
49 import matplotlib.pyplot as plt
50 from PIL import Image
51
52 # Create a simple test image
53 width, height = 400, 300
54 image = np.zeros((height, width, 3), dtype=np.uint8)
55
56 # Add some patterns
```

```

57 image[:, :width//3, 0] = 255 # Red section
58 image[:, width//3:2*width//3, 1] = 255 # Green section
59 image[:, 2*width//3:, 2] = 255 # Blue section
60
61 # Add a white circle in the center
62 center_x, center_y = width//2, height//2
63 radius = 50
64 Y, X = np.ogrid[:height, :width]
65 mask = (X - center_x)**2 + (Y - center_y)**2 <= radius**2
66 image[mask] = [255, 255, 255]
67
68 # Display and save
69 plt.figure(figsize=(8, 6))
70 plt.imshow(image)
71 plt.title('Test Image with RGB Stripes and White Circle')
72 plt.axis('off')
73 plt.savefig('test_image.png', dpi=150, bbox_inches='tight')
74 plt.show()
75
76 print(f"Created test image: {width}x{height} pixels")
77 print(f"Image shape: {image.shape}")
78 print(f"Image dtype: {image.dtype}")

```

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)

Gunakan perintah |includegraphics untuk menyisipkan gambar

4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

1. **Mengapa penting menggunakan environment terpisah untuk project multimedia?**
Karena setiap project memiliki ketentuan dan kebutuhan library yang berbeda-beda, maka dari itu kita harus memisahkan setiap project dengan environment yang berbeda-beda
2. **Apa perbedaan utama antara conda, venv, dan uv? Mengapa Anda memilih tool yang Anda gunakan?**
Karena uv itu yang paling cepat dan modern, seperti bun jika di kasus node.js
3. **Library mana yang paling sulit diinstall dan mengapa?**
semuanya mudah, jika cepat membaca dan memahami sebuah dokumentasi
4. **Bagaimana cara mengatasi masalah dependency conflict jika terjadi?**
cek versi, stackoverflow, chatgpt, atau AI lainnya
5. **Jelaskan fungsi dari masing-masing library yang berhasil Anda install!**

Fungsi Library yang Diinstall:

- **Library Audio Processing**

- **librosa**: Digunakan untuk analisis dan pemrosesan sinyal audio, khususnya musik, seperti ekstraksi fitur spectrogram, chroma, tempo, dan beat tracking.
- **soundfile**: Untuk membaca dan menulis file audio (misalnya WAV, FLAC) dengan mudah.
- **scipy**: Pustaka saintifik umum yang memiliki modul pemrosesan sinyal, filtering, dan analisis data audio.

- **Library Image Processing**

- **opencv (cv2)**: Pustaka visi komputer populer untuk pemrosesan gambar, deteksi wajah, segmentasi, dan analisis citra.
- **pillow (PIL)**: Untuk membuka, memodifikasi, dan menyimpan berbagai format gambar.
- **scikit-image**: Menyediakan algoritma pemrosesan citra berbasis NumPy dan SciPy, seperti segmentasi, denoising, dan feature extraction.
- **matplotlib**: Library visualisasi data yang dapat digunakan untuk menampilkan gambar atau hasil pengolahan citra dalam bentuk grafik.

- **Library Video Processing**

- **ffmpeg**: Perangkat lunak open-source untuk pengolahan video dan audio, termasuk konversi format, kompresi, dan ekstraksi frame.
- **moviepy**: Library Python berbasis ffmpeg untuk mengedit video, memotong, menggabungkan, menambahkan teks, efek, atau audio.

- **Library General Purpose**

- **numpy**: Fondasi komputasi numerik di Python, menyediakan array multidimensi dan operasi matematis cepat.
- **pandas**: Untuk analisis data terstruktur dalam bentuk tabel (dataframe), mirip Excel tetapi lebih fleksibel.
- **jupyter**: Lingkungan notebook interaktif untuk menulis kode, dokumentasi, visualisasi, dan eksekusi secara langsung.

4.4 Troubleshooting

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

- **Masalah 1: Error saat menggunakan which python**

Solusi: copy paste eror ke chatgpt, lalu mencoba lagi dengan solusi yang didapatkan

- **Masalah 2: Saat menggunakan pip dengan kasus uv itu lemot installation nya**

Solusi: setelah baca dokumentasi baru tahu bahwa saat installation package uv bisa dipakai di sebelum pip install dengan contoh uv pip install <nama package>

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 11: Export conda environment

5.2 Untuk venv/uv

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

Kode 12: 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 babel==2.17.0
9 beautifulsoup4==4.13.5
10 bleach==6.2.0
11 board==1.0
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 debugpy==1.8.16
18 decorator==5.2.1
19 defusedxml==0.7.1
20 executing==2.2.0
21 fastjsonschema==2.21.2
22 fqdn==1.5.1
23 h11==0.16.0
24 httpcore==1.0.9
25 httpx==0.28.1
26 idna==3.10
27 ipykernel==6.30.1
28 ipython==9.5.0
29 ipython_pygments_lexers==1.1.1
30 ipywidgets==8.1.7
31 isoduration==20.11.0
32 jedi==0.19.2
33 Jinja2==3.1.6
34 json5==0.12.1
35 jsonpointer==3.0.0
36 jsonschema==4.25.1
37 jsonschema-specifications==2025.4.1
38 jupyter==1.1.1
39 jupyter-console==6.6.3
40 jupyter-events==0.12.0
41 jupyter-lsp==2.3.0
42 jupyter_client==8.6.3
43 jupyter_core==5.8.1
44 jupyter_server==2.17.0
45 jupyter_server_terminals==0.5.3
46 jupyterlab==4.4.6
47 jupyterlab_pygments==0.3.0
48 jupyterlab_server==2.27.3
49 jupyterlab_widgets==3.0.15
50 lark==1.2.2
51 m3u8==6.0.0
52 MarkupSafe==3.0.2
53 matplotlib-inline==0.1.7
54 mistune==3.1.4
55 nbclient==0.10.2
```

```

56 nbconvert==7.16.6
57 nbformat==5.10.4
58 nest-asyncio==1.6.0
59 notebook==7.4.5
60 notebook_shim==0.2.4
61 numpy==2.3.2
62 packaging==25.0
63 pandas==2.3.2
64 pandocfilters==1.5.1
65 parso==0.8.5
66 pillow==11.3.0
67 platformdirs==4.4.0
68 prometheus_client==0.22.1
69 prompt_toolkit==3.0.52
70 psutil==7.0.0
71 pure_eval==0.2.3
72 pycparser==2.22
73 Pygments==2.19.2
74 python-dateutil==2.9.0.post0
75 python-json-logger==3.3.0
76 pytz==2025.2
77 pywin32==311
78 pywintypes==3.0.0
79 PyYAML==6.0.2
80 pyzmq==27.0.2
81 referencing==0.36.2
82 requests==2.32.5
83 rfc3339-validator==0.1.4
84 rfc3986-validator==0.1.1
85 rfc3987-syntax==1.1.0
86 rpds-py==0.27.1
87 Send2Trash==1.8.3
88 setuptools==80.9.0
89 six==1.17.0
90 sniffio==1.3.1
91 soupsieve==2.8
92 stack-data==0.6.3
93 terminado==0.18.1
94 tinycss2==1.4.0
95 tornado==6.5.2
96 traitlets==5.14.3
97 types-python-dateutil==2.9.0.20250822
98 typing_extensions==4.15.0
99 tzdata==2025.2
100 uri-template==1.3.0
101 urllib3==2.5.0
102 wcwidth==0.2.13
103 webcolors==24.11.1
104 webencodings==0.5.1
105 websocket-client==1.8.0
106 widgetsnbextension==4.0.14

```

Kode 13: 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

Jawaban:

- Proses setup Python environment untuk multimedia berhasil dilakukan dengan menggunakan **uv** sebagai tool manajemen environment. Pemilihan ini didasarkan pada kecepatan dan sifatnya yang modern, sehingga mempermudah pengelolaan dependency dan meminimalkan konflik antar library.
- Instalasi library multimedia yang meliputi audio, image, video, dan general purpose dapat dijalankan dengan lancar. Library yang digunakan (seperti *librosa*, *opencv*, *moviepy*, *numpy*, dan lain-lain) berhasil terpasang dengan benar serta telah diuji melalui skrip sederhana untuk memverifikasi fungsionalitasnya.
- Hasil uji coba berupa plot gelombang sine (audio) dan pembuatan citra sederhana (image) menunjukkan bahwa environment dan library yang diinstall bekerja sesuai harapan. Proses dokumentasi dalam L^AT_EX juga membantu memastikan setiap langkah dapat direproduksi dengan baik.
- Troubleshooting yang muncul, seperti penggunaan perintah **which** di Windows, dapat diatasi dengan menggantinya menggunakan **where**, sesuai dengan perintah yang berlaku pada sistem operasi Windows. Selain itu, pemahaman dokumentasi resmi juga sangat membantu dalam mengatasi kendala instalasi.
- Dengan adanya environment yang terpisah, setiap project multimedia dapat dikembangkan secara lebih aman dan terstruktur tanpa risiko benturan versi library dengan project lain. Hal ini mempersiapkan mahasiswa untuk mengerjakan project multimedia selanjutnya secara lebih profesional.
- Untuk mahasiswa lain yang akan melakukan setup serupa, disarankan untuk:
 - Membaca dokumentasi resmi setiap library sebelum instalasi.
 - Menggunakan manajemen environment seperti **conda**, **venv**, atau **uv** untuk menghindari dependency conflict.
 - Mendokumentasikan setiap langkah instalasi agar proses dapat direproduksi di kemudian hari.

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

- <https://chatgpt.com/share/e/68b1d146-8104-8003-aa8e-b16e88c36f2f>

References