



Nama: **Harisya Miranti (122140049)** 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 : uv**
- Screenshot atau copy-paste output dari perintah verifikasi environment

```
C:\Multimedia\tugas1>pip install uv
Requirement already satisfied: uv in c:\users\user\appdata\local\programs\python\python312\lib\
site-packages (0.8.13)

[notice] A new release of pip is available: 25.1.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Multimedia\tugas1>mulmed-uv\Scripts\activate

(mulmed-uv) C:\Multimedia\tugas1>where python
C:\Multimedia\tugas1\mulmed-uv\Scripts\python.exe
C:\Users\User\AppData\Local\Programs\Python\Python312\python.exe
C:\Users\User\AppData\Local\Microsoft\WindowsApps\python.exe
```

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

3.4 Bagian 3: Verifikasi Instalasi

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

```

1  # test_libs.py
2
3  print("=== TEST GENERAL LIBRARIES ===")
4  try:
5      import numpy as np
6      import pandas as pd
7      print("NumPy version:", np.__version__)
8      print("Pandas version:", pd.__version__)
9  except Exception as e:
10     print("General libs error:", e)
11
12  print("\n=== TEST AUDIO LIBRARIES ===")
13  try:
14      import librosa
15      import soundfile as sf
16      import scipy
17      print("Librosa version:", librosa.__version__)
18      print("SoundFile version:", sf.__version__)
19      print("Scipy version:", scipy.__version__)
20  except Exception as e:
21     print("Audio libs error:", e)
22
23  print("\n=== TEST IMAGE LIBRARIES ===")
24  try:
25      import cv2
26      from PIL import Image
27      import skimage
28      import matplotlib
29      print("OpenCV version:", cv2.__version__)
30      print("Pillow version:", Image.__version__)
31      print("scikit-image version:", skimage.__version__)
32      print("Matplotlib version:", matplotlib.__version__)
33  except Exception as e:
34     print("Image libs error:", e)
35
36  print("\n=== TEST VIDEO LIBRARY ===")
37  try:
38      import moviepy
39      print("MoviePy version:", moviepy.__version__)
40  except Exception as e:
41     print("Video lib error:", e)
42
43  print("\n Semua library berhasil di-import tanpa error!")
44  scikit-image matplotlib
45

```

Kode 8: Instalasi library image

Jalankan script dan dokumentasikan hasilnya:

```
=== TEST GENERAL LIBRARIES ===
NumPy version: 2.2.6
Pandas version: 2.3.2

=== TEST AUDIO LIBRARIES ===
Librosa version: 0.11.0
SoundFile version: 0.13.1
Scipy version: 1.16.1

=== TEST IMAGE LIBRARIES ===
OpenCV version: 4.12.0
Pillow version: 11.3.0
scikit-image version: 0.25.2
Matplotlib version: 3.10.5

=== TEST VIDEO LIBRARY ===
MoviePy version: 2.1.2

✅ Semua library berhasil di-import tanpa error!
```

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 9: 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 10: 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 [PASTE OUTPUT DI SINI]
```

Kode 11: Output verifikasi instalasi

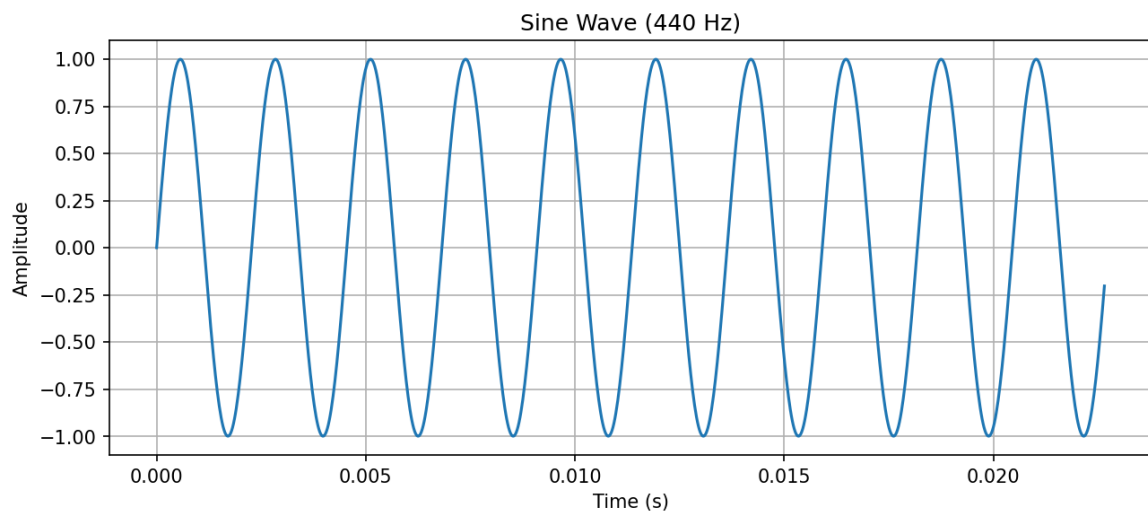
4.2 Screenshot Hasil Test

Sisipkan screenshot atau gambar hasil dari:

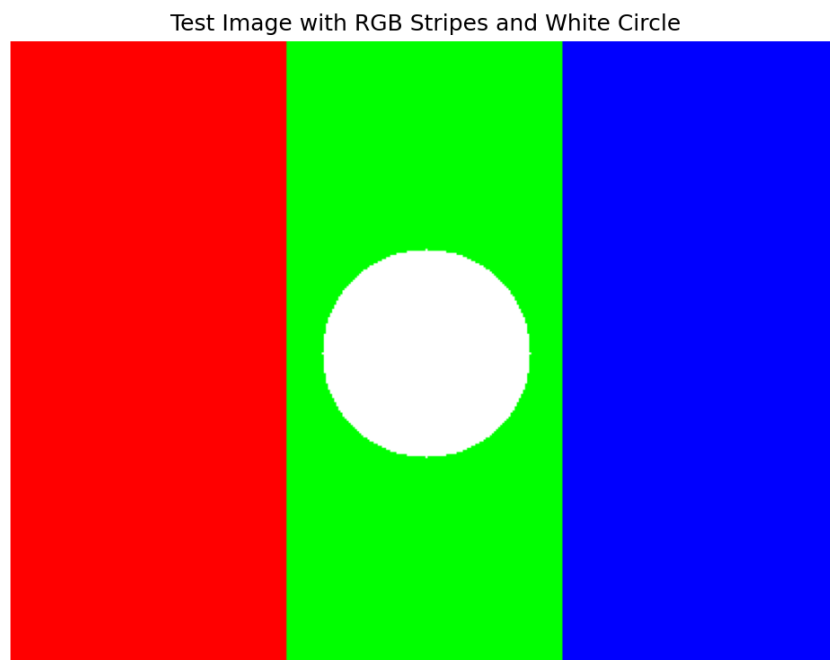
- Terminal/command prompt yang menunjukkan environment aktif

```
=== TEST GENERAL LIBRARIES ===  
NumPy version: 2.2.6  
Pandas version: 2.3.2  
  
=== TEST AUDIO LIBRARIES ===  
Librosa version: 0.11.0  
SoundFile version: 0.13.1  
Scipy version: 1.16.1  
  
=== TEST IMAGE LIBRARIES ===  
OpenCV version: 4.12.0  
Pillow version: 11.3.0  
scikit-image version: 0.25.2  
Matplotlib version: 3.10.5  
  
=== TEST VIDEO LIBRARY ===  
MoviePy version: 2.1.2  
  
✅ Semua library berhasil di-import tanpa error!
```

- 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 environment penting untuk membantu menghindari terjadinya konflik antar project yang sudah dibuat, sehingga project dapat digunakan tanpa saling bentrok.
- 2. Apa perbedaan utama antara conda, venv, dan uv? Mengapa Anda memilih tool yang Anda gunakan?**
kalau conda lebih cocok untuk pengguna pemula sehingga lebih mudah untuk digunakan, kalau venv itu bawaan dari python jadi tidak perlu install tambahan, kalau uv lebih modern dan cepat. saya memilih uv karena uv kompatibel dengan ekosistem python dan praktis untuk project multimedia.
- 3. Library mana yang paling sulit diinstall dan mengapa?**
menurut pendapat saya yang paling sulit diinstall adalah conda, karena dilihat dari step yang banyak dan ukuran file yang besar.
- 4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?**
dengan cara selalu menggunakan virtual environment dan gunakan versi library yang paling cocok.
- 5. Jelaskan fungsi dari masing-masing library yang berhasil Anda install!**
library librosa, soundfile, dan scipy untuk mengolah file audio, opencv-python, pillow, scikit-image, matplotlib untuk mengolah file gambar, dan library moviepy berfungsi untuk mengolah file video.

4.4 Troubleshooting

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

- **Masalah 1:** saat menginstall dependencies terjadi masalah konektivitas sehingga instal dependencies sering gagal dan berulang terus.

Solusi: menggunakan jaringan hotspot pribadi yang lebih kencang konektivitasnya

- **Masalah 2:** salah memasukkan versi python saat ingin membuat virtual environment, yang harusnya versi 3.10, saya menggunakan versi 3.12.3

Solusi: membuat ulang virtual environment sesuai dengan versi python yang disarankan, yaitu versi 3.10

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 12: Export conda environment

5.2 Untuk venv/uv

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

Kode 13: 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==1.3.3
18 cycler==0.12.1
19 debugpy==1.8.16
20 decorator==5.2.1
21 defusedxml==0.7.1
22 executing==2.2.0
23 fastjsonschema==2.21.2
24 fonttools==4.59.2
25 fqdn==1.5.1
26 h11==0.16.0
27 httpcore==1.0.9
28 httpx==0.28.1
29 idna==3.10
30 imageio==2.37.0
31 imageio-ffmpeg==0.6.0
```

```
32 ipykernel==6.30.1
33 ipython==9.5.0
34 ipython-pygments-lexers==1.1.1
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==3.1.4
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==3.5
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 packaging==25.0
78 pandas==2.3.2
79 pandocfilters==1.5.1
80 parso==0.8.5
81 pillow==11.3.0
82 platformdirs==4.4.0
83 pooch==1.8.2
84 proglog==0.1.12
85 prometheus-client==0.22.1
86 prompt-toolkit==3.0.52
87 psutil==7.0.0
88 pure-eval==0.2.3
89 pycparser==2.22
90 pygments==2.19.2
91 pyparsing==3.2.3
92 python-dateutil==2.9.0.post0
93 python-dotenv==1.1.1
```

```
94 python-json-logger==3.3.0
95 pytz==2025.2
96 pywin32==311
97 pywinpty==3.0.0
98 pyyaml==6.0.2
99 pyzmq==27.0.2
100 referencing==0.36.2
101 requests==2.32.5
102 rfc3339-validator==0.1.4
103 rfc3986-validator==0.1.1
104 rfc3987-syntax==1.1.0
105 rpds-py==0.27.1
106 scikit-image==0.25.2
107 scikit-learn==1.7.1
108 scipy==1.16.1
109 send2trash==1.8.3
110 setuptools==80.9.0
111 six==1.17.0
112 sniffio==1.3.1
113 soundfile==0.13.1
114 soupsieve==2.8
115 soxr==0.5.0.post1
116 stack-data==0.6.3
117 terminado==0.18.1
118 threadpoolctl==3.6.0
119 tiffiff==2025.8.28
120 tinycss2==1.4.0
121 tornado==6.5.2
122 tqdm==4.67.1
123 traitlets==5.14.3
124 types-python-dateutil==2.9.0.20250822
125 typing-extensions==4.15.0
126 tzdata==2025.2
127 uri-template==1.3.0
128 urllib3==2.5.0
129 wcwidth==0.2.13
130 webcolors==24.11.1
131 webencodings==0.5.1
132 websocket-client==1.8.0
133 widgetsnbextension==4.0.14
134 ]
```

Kode 14: 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

pengetahuan baru tentang instalansi environment python yang akan digunakan dalam pembelajaran multimedia, dan cara menangani trouble yang dihadapi saat proses penginstalan yang dibantu oleh AI dan teman yang sudah berhasil menginstal dukan. Persiapan untuk project multimedia selanjutnya adalah mulai mencari tahu terkait library yang akan digunakan seperti library audio, video, dan gambar. saran untuk teman mahasiswa lainnya adalah jangan ragu untuk menanyakan kendala yang dihadapi ke teman yang sudah tahu dan berhasil melakukan setup.

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

References

chatGPT: <https://chatgpt.com/share/68b1b8df-81f8-8013-8f68-e76e33994c77> rekan sejawat: Elma Nurul fatika