



Nama: **Ferdana Al-Hakim (122140012)**
Environment untuk Multimedia

Tugas Ke: **Worksheet 1: Setup Python**

Mata Kuliah: **Sistem Teknologi Multimedia (IF25-40305)**

Tanggal: August 28, 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 \LaTeX

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 \LaTeX 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: **UV**
- Screenshot atau copy-paste output dari perintah verifikasi environment

```
1 (multimedia-uv) PS C:\KULIAH\Semester 7\Sistem Teknologi Multimedia> where python
2 C:\KULIAH\Semester 7\Sistem Teknologi Multimedia\multimedia-uv\Scripts\python.exe
3 C:\Users\AppData\Local\Programs\Python\Python311\python.exe
4
5 (multimedia-uv) PS C:\KULIAH\Semester 7\Sistem Teknologi Multimedia> uv --version
6 uv 0.5.14 (Cargo 1.82.0 2024-10-12)
7
```

Kode 4: Output verifikasi environment 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 5: 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 6: 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 7: 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 8: Instalasi library umum

Dokumentasikan di sini:

- Perintah instalasi yang Anda gunakan: `uv pip install -r requirement.txt`

- Screenshot proses instalasi atau output sukses: Terlampir di Figure/env.png
- Daftar library yang berhasil diinstall dengan versinya:
 - **Audio Processing:** librosa==0.11.0, soundfile==0.13.1, scipy==1.16.1
 - **Image Processing:** opencv-python==4.12.0.88, pillow==11.3.0, scikit-image==0.25.2, matplotlib==3.10.5
 - **Video Processing:** moviepy==2.2.1, ffmpeg-python==0.2.0
 - **General Purpose:** numpy==2.2.6, pandas==2.3.2, jupyter==1.1.1

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 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 Verifikasi Manual Library

Untuk memverifikasi instalasi, dilakukan test import sederhana pada library utama dengan hasil sebagai berikut:

```

1 (multimedia-uv) PS C:\KULIAH\Semester 7\Sistem Teknologi Multimedia> python
2 Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
3 Type "help", "copyright", "credits" or "license" for more information.
4
5 >>> import numpy as np
6 >>> import pandas as pd
7 >>> import matplotlib.pyplot as plt
8 >>> import librosa
9 >>> import cv2
10 >>> from PIL import Image
11 >>> import moviepy.editor as mp
12 >>> print("Semua library berhasil diimport!")
13 Semua library berhasil diimport!
14 >>> exit()

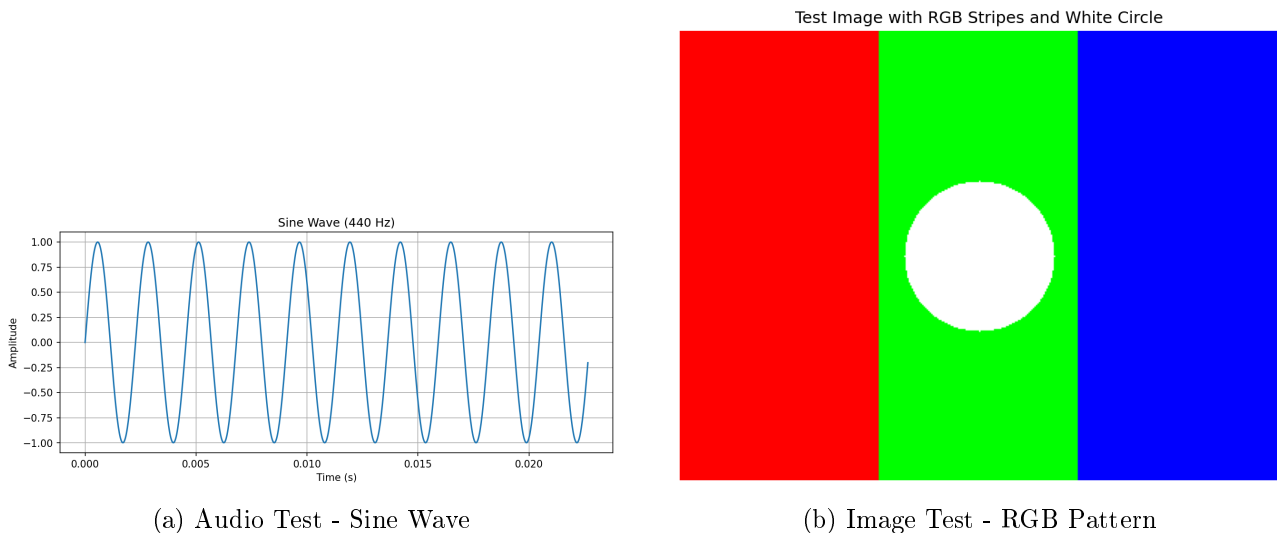
```

Kode 11: Output test import library multimedia

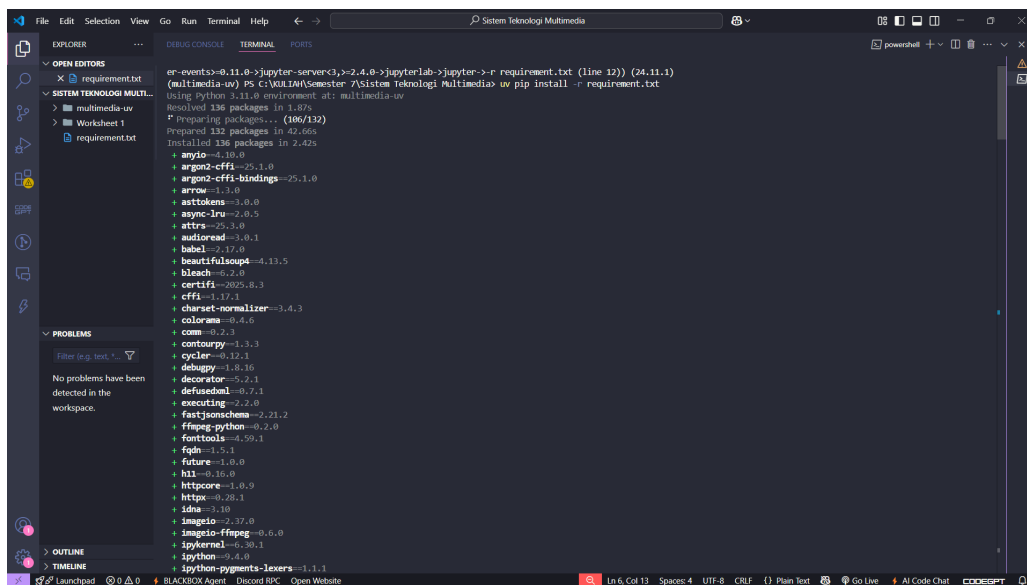
4.2 Screenshot Hasil Test

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)



Gambar 1: Hasil Test Simple Code untuk Audio dan Image Processing



Gambar 2: Screenshot Environment Setup dan Import Test

4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

1. Mengapa penting menggunakan environment terpisah untuk project multimedia?

Environment terpisah sangat krusial untuk project multimedia karena library multimedia sering memiliki dependency yang kompleks dan spesifik. Misalnya, OpenCV membutuhkan versi NumPy tertentu, sedangkan librosa mungkin butuh versi berbeda. Tanpa environment terpisah, kita bisa mengalami konflik dependency yang membuat project tidak berjalan. Selain itu, multimedia project biasanya memerlukan library dengan ukuran besar seperti TensorFlow atau PyTorch, sehingga isolasi environment membantu menjaga sistem tetap bersih.

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

Conda adalah package manager yang bisa mengelola binary dependencies, cocok untuk data science. **Venv** adalah built-in Python yang sederhana tapi hanya mengelola Python packages. **UV** adalah tool modern yang sangat cepat dalam resolving dependencies dan instalasi. Saya memilih UV karena kecepatannya yang luar biasa - instalasi library yang biasanya memakan waktu menit dengan pip bisa selesai dalam hitungan detik. Plus, UV memiliki dependency resolver yang lebih canggih sehingga mengurangi risiko konflik.

3. Library mana yang paling sulit diinstall dan mengapa?

MoviePy adalah library yang paling challenging karena memerlukan FFmpeg sebagai dependency eksternal. Di Windows, FFmpeg tidak otomatis terinstall dan perlu setup manual atau melalui package manager seperti chocolatey. Selain itu, OpenCV juga kadang bermasalah karena ada beberapa variant (opencv-python, opencv-contrib-python) yang bisa konflik. Librosa juga terkadang bermasalah di Windows karena dependency pada library audio backend seperti soundfile.

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

Langkah pertama adalah menggunakan **pip check** untuk mengidentifikasi konflik. Jika ada konflik, coba install versi spesifik yang kompatibel dengan **pip install package==version**. Alternatifnya, gunakan **pip-tools** atau **pipdeptree** untuk melihat dependency tree. Jika masih bermasalah, buat environment baru yang clean dan install library satu per satu sesuai prioritas. UV sangat membantu di sini karena dependency resolver-nya lebih pintar dalam menangani konflik dibanding pip biasa.

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

Audio: Librosa untuk analisis audio (spektogram, MFCC), SoundFile untuk I/O audio files, SciPy untuk signal processing. **Image:** OpenCV untuk computer vision dan image processing, Pillow untuk basic image operations, Scikit-image untuk advanced image processing algorithms. **Video:** MoviePy untuk editing dan manipulasi video, FFmpeg-python sebagai wrapper FFmpeg. **General:** NumPy untuk numerical computing, Pandas untuk data manipulation, Matplotlib untuk plotting dan visualization, Jupyter untuk interactive development environment.

4.4 Troubleshooting

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

- **Masalah 1:** MoviePy import error karena FFmpeg tidak ditemukan
Solusi: Install FFmpeg secara manual atau gunakan **conda install ffmpeg** jika menggunakan conda. Untuk Windows, bisa download FFmpeg binary dan tambahkan ke PATH.
- **Masalah 2:** Librosa instalasi lambat karena building dari source
Solusi: Gunakan pre-compiled wheel dengan **pip install -only-binary=all librosa** atau switch ke conda yang sudah menyediakan binary 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 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 librosa==0.11.0
2 soundfile==0.13.1
3 scipy==1.16.1
4 opencv-python==4.12.0.88
5 pillow==11.3.0
6 scikit-image==0.25.2
7 matplotlib==3.10.5
8 moviepy==2.2.1
9 ffmpeg-python==0.2.0
10 numpy==2.2.6
11 pandas==2.3.2
12 jupyter==1.1.1
```

Kode 14: Requirements.txt untuk UV Environment

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

Pengalaman setup Python environment untuk multimedia cukup menantang karena kompleksitas dependency antar library. UV terbukti sangat membantu dengan kecepatan instalasi yang luar biasa dan dependency resolution yang lebih baik dibanding pip. Environment yang sudah disiapkan ini memberikan foundation yang solid untuk project multimedia selanjutnya, dengan semua library utama (audio, image, video processing) sudah terintegrasi dengan baik.

Untuk mahasiswa lain, saya sarankan: 1) Selalu gunakan environment terpisah untuk setiap project, 2) Pertimbangkan UV untuk kecepatan dan reliability, 3) Test instalasi dengan script sederhana seperti yang sudah dibuat, 4) Dokumentasikan versi library yang digunakan untuk reproducibility, 5) Siapkan requirements.txt dari awal untuk memudahkan sharing dan deployment project.

7 Referensi

1. LLM GPT

<https://chatgpt.com/share/68aed053-1fec-800f-9a49-b502ec5eb92e>