



Nama: **Dina Rahma Dita (122140184)** Tugas Ke: **Worksheet 1: Setup Python Environment untuk Multimedia**

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

Tanggal: September 13, 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: **[Conda]**
- Screenshot atau copy-paste output dari perintah verifikasi environment

```

Anaconda Prompt (Anaconda) x + v
(base) C:\Users\Dina Rahma Dita>conda create -n multimedia python=3.11
3 channel Terms of Service accepted
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Anaconda3\envs\multimedia

  added / updated specs:
    - python=3.11

The following NEW packages will be INSTALLED:

bzip2                pkgs/main/win-64::bzip2-1.0.8-h2bbff1b_6
ca-certificates      pkgs/main/win-64::ca-certificates-2025.9.9-haa95532_0
expat                pkgs/main/win-64::expat-2.7.1-h8ddb27b_0
libffi               pkgs/main/win-64::libffi-3.4.4-hd77b12b_1
libzlib              pkgs/main/win-64::libzlib-1.3.1-h02ab6af_0
openssl              pkgs/main/win-64::openssl-3.0.17-h35632f6_0
pip                  pkgs/main/noarch::pip-25.2-pyh872135_0
python               pkgs/main/win-64::python-3.11.13-h981015d_0
setuptools           pkgs/main/win-64::setuptools-78.1.1-py311haa95532_0
sqlite               pkgs/main/win-64::sqlite-3.50.2-hds9a48d_1
tk                   pkgs/main/win-64::tk-8.6.15-hf190647_0
tzdata               pkgs/main/noarch::tzdata-2025b-h04d1e81_0
ucrt                  pkgs/main/win-64::ucrt-10.0.22621.0-haa95532_0
vc                    pkgs/main/win-64::vc-14.3-h2df5915_10
vc14_runtime         pkgs/main/win-64::vc14_runtime-14.44.35208-h4927774_10
vs2015_runtime       pkgs/main/win-64::vs2015_runtime-14.44.35208-ha6b5a95_10
wheel                pkgs/main/win-64::wheel-0.45.1-py311haa95532_0
xz                   pkgs/main/win-64::xz-5.6.4-h4754444_1
zlib                 pkgs/main/win-64::zlib-1.3.1-h02ab6af_0

Proceed ([y]/n)? y

Downloading and Extracting Packages:
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate multimedia
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) C:\Users\Dina Rahma Dita>

(base) C:\Users\Dina Rahma Dita>conda activate multimedia
(multimedia) C:\Users\Dina Rahma Dita>conda info --envs

# conda environments:
#
base                C:\Anaconda3
multimedia           * C:\Anaconda3\envs\multimedia

(multimedia) C:\Users\Dina Rahma Dita>

```

Gambar 1: Hasil verifikasi environment dengan conda

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

```

Kode 4: Instalasi library audio

```
(multimedia) C:\Users\Dina Rahma Dita>conda install -c conda-forge librosa pysoundfile scipy
3 channel Terms of Service accepted
Channels:
- conda-forge
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Anaconda3\envs\multimedia

added / updated specs:
- librosa
- pysoundfile
- scipy

The following packages will be downloaded:
```

package	build	size	channel
_openmp_mutex-4.5	2.gnu	48 KB	conda-forge
audioread-3.0.1	py311h1ea47a8_2	44 KB	conda-forge
brotili-1.1.0	hfd05255_4	20 KB	conda-forge
brotili-bin-1.1.0	hfd05255_4	21 KB	conda-forge
contourpy-1.3.3	py311h3f6045d_2	220 KB	conda-forge
cycler-0.12.1	pyhd8ed1ab_1	13 KB	conda-forge
fonttools-4.59.2	py311h3f79411_0	2.4 MB	conda-forge
freetype-2.14.0	h57928b3_1	180 KB	conda-forge
joblib-1.5.2	pyhd8ed1ab_0	219 KB	conda-forge
kiwisolver-1.4.9	py311h275cad7_1	72 KB	conda-forge
lazy_loader-0.4	pyhd8ed1ab_2	7 KB	conda-forge
lcms2-2.17	hbcbf6048_0	499 KB	conda-forge
lerc-4.0.0	h6470a55_1	161 KB	conda-forge
libblas-3.9.0	35_h5709861_mkl	64 KB	conda-forge
libbrotlicommon-1.1.0	hfd05255_4	70 KB	conda-forge
libbrotlidec-1.1.0	hfd05255_4	33 KB	conda-forge
libbrotlienc-1.1.0	hfd05255_4	240 KB	conda-forge
libcbblas-3.9.0	35_h2a3cdd5_mkl	65 KB	conda-forge
libdeflate-1.22	h2466b09_0	152 KB	conda-forge
libflac-1.4.3	h63175ca_0	323 KB	conda-forge
libfreetype-2.14.0	h57928b3_1	8 KB	conda-forge
libfreetype6-2.14.0	hdbae1cb_1	332 KB	conda-forge
libgcc-15.1.0	h1383e82_5	653 KB	conda-forge
libgomp-15.1.0	h1383e82_5	523 KB	conda-forge
libhwloc-2.12.1	default_h88281d1_1000	2.3 MB	conda-forge

```

The following packages will be UPDATED:

openssl                                pkgs/main::openssl-3.0.17-h35632f6_0 --> conda-forge::openssl-3.5.2-h725018a_0

Proceed ([y]/n)? y

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(multimedia) C:\Users\Dina Rahma Dita>
```

Gambar 2: Tampilan proses instalasi library audio di terminal

3.3.2 Library Image Processing

```
1 # Untuk conda:
2 conda install -c conda-forge opencv pillow scikit-image matplotlib
```

Kode 5: Instalasi library image

```
(multimedia) C:\Users\Dina Rahma Dita>conda install -c conda-forge opencv pillow scikit-image matplotlib
3 channel Terms of Service accepted
Channels:
- conda-forge
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Anaconda3\envs\multimedia

added / updated specs:
- matplotlib
- opencv
- pillow
- scikit-image

The following packages will be downloaded:
```

package	build	size	channel
_libavif-api-1.1.1	h57928b3_3	11 KB	conda-forge
aom-3.9.1	he0c23c2_0	1.9 MB	conda-forge
blosc-1.21.6	h85f69ea_0	49 KB	conda-forge
c-blosc2-2.15.2	hb461149_0	210 KB	conda-forge
cairo-1.18.4	h5782bbf_0	1.5 MB	conda-forge
charls-2.4.2	h1537add_0	94 KB	conda-forge
david-1.2.1	hcfcfb64_0	604 KB	conda-forge
double-conversion-3.3.1	he0c23c2_0	70 KB	conda-forge
ffmpeg-7.1.0	gpl_h2585aa8_705	9.5 MB	conda-forge
font-ttf-dejavu-sans-mono-2.37	hab24e00_0	388 KB	conda-forge
font-ttf-inconsolata-3.000	h77eed37_0	94 KB	conda-forge
font-ttf-source-code-pro-2.038	h77eed37_0	684 KB	conda-forge
font-ttf-ubuntu-0.83	h77eed37_3	1.5 MB	conda-forge
fontconfig-2.15.0	h765892d_1	188 KB	conda-forge
fonts-conda-ecosystem-1	0	4 KB	conda-forge
fonts-conda-forge-1	0	4 KB	conda-forge
freelut-3.2.2	he0c23c2_3	109 KB	conda-forge
fribidi-1.0.16	hfd05255_0	63 KB	conda-forge
gdk-pixbuf-2.42.12	hed59a49_0	512 KB	conda-forge
giflib-5.2.2	h64bf75a_0	83 KB	conda-forge
harfbuzz-11.0.0	h9e37d49_0	1.1 MB	conda-forge
hdf5-1.14.3	nompi_hb2c4d47_109	2.0 MB	conda-forge
icu-75.1	he0c23c2_0	13.9 MB	conda-forge
imagecodecs-2024.9.22	py311h58741bd_0	1.6 MB	conda-forge
imageio-2.37.0	pyhf79c49_0	286 KB	conda-forge
imath-3.1.12	hbb528cf_0	157 KB	conda-forge

```

Proceed ([y]/n)? y

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: |
|
done

(multimedia) C:\Users\Dina Rahma Dita>

```

Gambar 3: Tampilan proses instalasi library image processing di terminal

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

```

(multimedia) C:\Users\Dina Rahma Dita>conda install -c conda-forge ffmpeg
3 channel Terms of Service accepted
Channels:
- conda-forge
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Anaconda3\envs\multimedia

  added / updated specs:
  - ffmpeg

The following NEW packages will be INSTALLED:

  ffmpeg           conda-forge/win-64::ffmpeg-7.1.0-gpl_h2585aa8_705
  fribidi           conda-forge/win-64::fribidi-1.0.16-hfd85255_0
  gdk-pixbuf        conda-forge/win-64::gdk-pixbuf-2.42.12-hed59a49_0
  librsvg           conda-forge/win-64::librsvg-2.58.4-h5ce5fed_3
 openh264          conda-forge/win-64::openh264-2.5.0-ha9db3cd_0
  pangocairo        conda-forge/win-64::pangocairo-1.50.3-h0c53d3b_1
  pangox264         conda-forge/win-64::x264-11164_3095-h8ffe710_2
  x265              conda-forge/win-64::x265-3.5-h2d74725_3

Proceed [[y]/n]? y

Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: /
|
done

(multimedia) C:\Users\Dina Rahma Dita>

(multimedia) C:\Users\Dina Rahma Dita>pip install moviepy
Collecting moviepy
  Using cached moviepy-2.2.1-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: decorator<6.0,>=4.0.2 in c:\users\dina rahma dita\appdata\roaming\python\python311\site-packages (from moviepy) (5.2.1)
Requirement already satisfied: imageio<3.0,>=2.5 in c:\anaconda3\envs\multimedia\lib\site-packages (from moviepy) (2.37.0)
Collecting imageio_ffmpeg>=0.2.0 (from moviepy)
  Using cached imageio_ffmpeg-0.6.0-py3-none-win_amd64.whl.metadata (1.5 kB)
Requirement already satisfied: numpy>=1.25.0 in c:\anaconda3\envs\multimedia\lib\site-packages (from moviepy) (2.2.6)
Collecting proglog<1.0.0 (from moviepy)
  Using cached proglog-0.1.12-py3-none-any.whl.metadata (794 bytes)
Collecting python-dotenv>=0.10 (from moviepy)
  Downloading python_dotenv-1.1.1-py3-none-any.whl.metadata (24 kB)
Requirement already satisfied: pillow<12.0,>=9.2.0 in c:\anaconda3\envs\multimedia\lib\site-packages (from moviepy) (11.3.0)
Collecting tqdm (from proglog<1.0.0->moviepy)
  Downloading tqdm-4.67.1-py3-none-any.whl.metadata (57 kB)
Requirement already satisfied: colorama in c:\users\dina rahma dita\appdata\roaming\python\python311\site-packages (from tqdm->proglog<1.0.0->moviepy) (0.4.6)
Using cached moviepy-2.2.1-py3-none-any.whl (129 kB)
Using cached proglog-0.1.12-py3-none-any.whl (6.3 kB)
Using cached imageio_ffmpeg-0.6.0-py3-none-win_amd64.whl (31.2 MB)
Downloading python_dotenv-1.1.1-py3-none-any.whl (20 kB)
Downloading tqdm-4.67.1-py3-none-any.whl (78 kB)
Installing collected packages: tqdm, python-dotenv, imageio_ffmpeg, proglog, moviepy
Successfully installed imageio_ffmpeg-0.6.0 moviepy-2.2.1 proglog-0.1.12 python-dotenv-1.1.1 tqdm-4.67.1

(multimedia) C:\Users\Dina Rahma Dita>

```

Gambar 4: Tampilan proses instalasi library video processing di terminal

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

3.4 Bagian 3: Verifikasi Instalasi

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

Jalankan script dan dokumentasikan hasilnya:

```
(multimedia) C:\Users\Dina Rahma Dita\Downloads\Worksheet 1\Worksheet 1>python verifikasi.py
=== Verifikasi Library ===
Python      : 3.11.13
pysoundfile : 0.13.1
OpenCV      : 4.12.0
Pillow      : 11.3.0
scikit-image: 0.25.2
matplotlib  : 3.10.6
moviepy     : 2.1.2
numpy       : 2.2.6
pandas      : 2.3.2
jupyter/notebook: 7.4.5
librosa     : 0.11.0
scipy       : 1.16.2

=== Verifikasi ffmpeg (imageio-ffmpeg) ===
Path : C:\Anaconda3\envs\multimedia\Lib\site-packages\imageio_ffmpeg\binaries\ffmpeg-win-x86_64-v7.1.exe
Info : 7.1-essentials_build-www.gyan.dev

(multimedia) C:\Users\Dina Rahma Dita\Downloads\Worksheet 1\Worksheet 1>
```

Gambar 6: Tampilan hasil verifikasi instalasi library

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
```



```

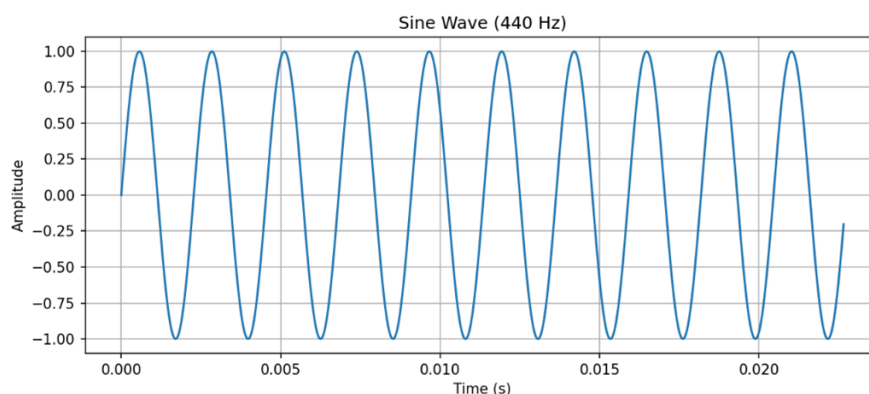
5 # Create a simple test image
6 width, height = 400, 300
7 image = np.zeros((height, width, 3), dtype=np.uint8)
8
9 # Add some patterns
10 image[:, :width//3, 0] = 255 # Red section
11 image[:, width//3:2*width//3, 1] = 255 # Green section
12 image[:, 2*width//3:, 2] = 255 # Blue section
13
14 # Add 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

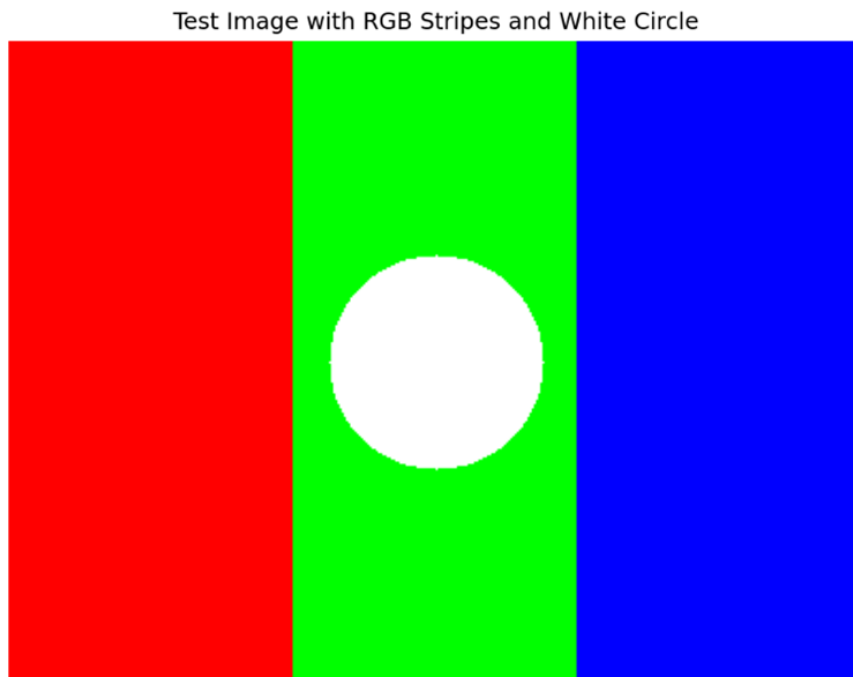
Dokumentasikan hasil eksekusi:

- 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

3.5.3 Test Audio Processing

Gambar 7: Tampilan hasil test audio processing

3.5.4 Test Image Processing



Gambar 8: Tampilan hasil test image processing

4 Bagian Laporan

4.1 Output Verifikasi Instalasi

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

```
(multimedia) C:\Users\Dina Rahma Dita\Downloads\Worksheet 1\Worksheet 1>python verifikasi.py
=== Verifikasi Library ===
Python      : 3.11.13
pysoundfile : 0.13.1
OpenCV      : 4.12.0
Pillow      : 11.3.0
scikit-image: 0.25.2
matplotlib  : 3.10.6
moviepy     : 2.1.2
numpy       : 2.2.6
pandas      : 2.3.2
jupyter/notebook: 7.4.5
librosa     : 0.11.0
scipy       : 1.16.2

=== Verifikasi ffmpeg (imageio-ffmpeg) ===
Path : C:\Anaconda3\envs\multimedia\Lib\site-packages\imageio_ffmpeg\binaries\ffmpeg-win-x86_64-v7.1.exe
Info  : 7.1-essentials_build-www.gyan.dev

(multimedia) C:\Users\Dina Rahma Dita\Downloads\Worksheet 1\Worksheet 1>
```

Gambar 9: Output verifikasi instalasi library

4.2 Screenshot Hasil Test

Sisipkan screenshot atau gambar hasil dari:

- Terminal/command prompt yang menunjukkan environment aktif

```
(base) C:\Users\Dina Rahma Dita>conda activate multimedia
(multimedia) C:\Users\Dina Rahma Dita>conda info --envs
# conda environments:
#
base                C:\Anaconda3
multimedia           * C:\Anaconda3\envs\multimedia
(multimedia) C:\Users\Dina Rahma Dita>
```

Gambar 10: Tampilan environment aktif di terminal

- Output dari script test audio (sine wave plot)

```
(base) C:\Users\Dina Rahma Dita>python test_audio.py
Generated 2s sine wave at 440 Hz
Sample rate: 44100 Hz
Total samples: 88200
```

Gambar 11: Output sine wave plot dari test audio processing

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

```
(base) C:\Users\Dina Rahma Dita>python test_image.py
Created test image: 400x300 pixels
Image shape: (300, 400, 3)
Image dtype: uint8

(base) C:\Users\Dina Rahma Dita>|
```

Gambar 12: Output RGB stripes dengan circle dari test image processing

Gunakan perintah `\includegraphics` untuk menyisipkan gambar

4.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

1. Mengapa penting menggunakan environment terpisah untuk project multimedia?

[Dalam mengerjakan project multimedia biasanya memakai banyak software dengan versi yang berbeda, bahkan kadang lintas sistem operasi jika dikerjakan bersama tim. Supaya project tidak rusak atau bermasalah, penting untuk membuat sebuah environment. Dengan adanya environment yang sama, semua kebutuhan sudah diseragamkan sehingga project bisa lebih stabil dan aman untuk dikerjakan bersama dengan tim.]

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

[Perbedaan antara conda, venv dan uv :

Conda = Bisa digunakan untuk library eksternal selain python, bahkan lintas bahasa lumayan powerfull tapi cukup berat dijalankan. Cocok untuk project ringan

Venv = Digunakan untuk library pure python, tidak bisa untuk library eksternal non-python

Uv = Dapat menginstall package python dengan lebih cepat tapi uv belum sekomplit conda]

3. Library mana yang paling sulit diinstall dan mengapa?

[Tidak terlalu sulit untuk menginstall library yang diminta, hanya ada sedikit masalah seperti pada saat menginstall library audio, ternyata dalam penulisannya harus ditulis pysoundfile bukan soundfile dan beda soundfile dan pysoundfile itu adalah jika soundfile itu modulnya maka pysoundfile itu librarynya.]

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

[Ada beberapa cara untuk mengatasi dependency conflict berdasarkan yang sudah saya lakukan :
 - Untuk mencegah konflik dengan projek lain maka dibuatlah environment terpisah (multimedia)
 - Melakukan downgrade atau upgrade agar tetap kompatibel, gunakan satu channel saja seperti conda-forge pada saat proses penginstallan untuk menghindari konflik dan file rusak.
 - Mengatur channel priority supaya library atau paket yang di install sumbernya hanya dari conda-forge
 - Jika konflik sudah terlalu berat maka bahkanuat ulang environment dan install ulang library yang dibutuhkan]

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

librosa = analisis & ekstraksi fitur audio (MFCC, spectrogram).

scipy = fungsi matematis & scientific (filter, transformasi).

pysoundfile = baca/tulis file audio (WAV, FLAC, OGG).

OpenCV (cv2) = olah gambar & video (deteksi objek, transformasi).

Pillow (PIL) = manipulasi gambar sederhana (resize, crop, ubah format).

scikit-image = image processing scientific (segmentasi, edge detection).

matplotlib = visualisasi data/grafik, tampilkan gambar & waveform audio.

ffmpeg = backend encoding/decoding audio-video.

moviepy = editing video (cut, merge, tambah teks/suara).

numpy = komputasi numerik & array multidimensi.

pandas = olah data tabular/metadata multimedia.

jupyter/notebook = coding interaktif, dokumentasi, & visualisasi.

4.4 Troubleshooting

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

- **Masalah 1:** *[Terkendala saat proses menginstall library audio karena tidak bisa menemukan modul soundfile]*

Solusi: *[mengganti ke pysoundfile bukan soundfile pada saat proses instalasi library audio]*

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 10: Export conda environment

5.2 Untuk venv/uv

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

Kode 11: Export pip requirements

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

```
1 [name: multimedia
2 channels:
3   - conda-forge
4   - defaults
5 dependencies:
6   - _libavif_api=1.1.1=h57928b3_3
7   - _openmp_mutex=4.5=2_gnu
8   - anyio=4.7.0=py311haa95532_0
9   - aom=3.9.1=he0c23c2_0
10  - argon2-cffi=21.3.0=pyhd3eb1b0_0
11  - argon2-cffi-bindings=21.2.0=py311h827c3e9_1
12  - asttokens=3.0.0=py311haa95532_0
13  - async-lru=2.0.4=py311haa95532_0
14  - attrs=24.3.0=py311haa95532_0
15  - audioread=3.0.1=py311h1ea47a8_2
16  - babel=2.16.0=py311haa95532_0
17  - beautifulsoup4=4.13.5=py311haa95532_0
18  - blas=1.0=mkl
19  - bleach=6.2.0=py311haa95532_0
20  - blosc=1.21.6=h85f69ea_0
21  - bottleneck=1.4.2=py311h57dcf0c_0
22  - brotli=1.1.0=hfd05255_4
23  - brotli-bin=1.1.0=hfd05255_4
24  - brotli-python=1.1.0=py311h3e6a449_4
25  - bzip2=1.0.8=h2bbff1b_6
26  - c-blosc2=2.15.2=hb461149_0
27  - ca-certificates=2025.9.9=haa95532_0
28  - cairo=1.18.0=h91e5215_2
29  - certifi=2025.8.3=pyhd8ed1ab_0
30  - cffi=1.17.1=py311h3485c13_1
31  - charls=2.4.2=h1537add_0
32  - charset-normalizer=3.4.3=pyhd8ed1ab_0
33  - colorama=0.4.6=py311haa95532_0
34  - comm=0.2.1=py311haa95532_0
35  - contourpy=1.3.3=py311h3fd045d_2
36  - cycler=0.12.1=pyhd8ed1ab_1
37  - davi1d=1.2.1=hcfcfb64_0
38  - debugpy=1.8.16=py311h885b0b7_0
39  - decorator=5.2.1=pyhd8ed1ab_0
40  - defusedxml=0.7.1=pyhd3eb1b0_0
41  - double-conversion=3.3.1=he0c23c2_0
42  - executing=2.2.1=py311haa95532_0
43  - expat=2.7.1=h8ddb27b_0
44  - ffmpeg=6.1.1=hc79a5da_3
45  - font-ttf-dejavu-sans-mono=2.37=h77eed37_0
46  - font-ttf-inconsolata=3.000=h77eed37_0
47  - font-ttf-source-code-pro=2.038=h77eed37_0
48  - font-ttf-ubuntu=0.83=h77eed37_3
49  - fontconfig=2.15.0=h765892d_1
50  - fonts-conda-ecosystem=1=0
51  - fonts-conda-forge=1=0
52  - fonttools=4.59.2=py311h3f79411_0
53  - freeglut=3.2.2=he0c23c2_3
54  - freetype=2.14.0=h57928b3_1
55  - fridibi=1.0.16=hfd05255_0
```

```

56 - gdk-pixbuf=2.42.12=hed59a49_0
57 - giflib=5.2.2=h64bf75a_0
58 - glib=2.80.2=h0df6a38_0
59 - glib-tools=2.80.2=h2f9d560_0
60 - graphite2=1.3.14=hac47afa_2
61 - h11=0.16.0=py311haa95532_0
62 - h2=4.3.0=pyhcf101f3_0
63 - harfbuzz=10.2.0=he2f9f60_1
64 - hdf5=1.14.3=nompi_hb2c4d47_109
65 - hpack=4.1.0=pyhd8ed1ab_0
66 - httpcore=1.0.9=py311haa95532_0
67 - httpx=0.28.1=py311haa95532_0
68 - hyperframe=6.1.0=pyhd8ed1ab_0
69 - icc_rt=2022.1.0=h6049295_2
70 - icu=73.2=h63175ca_0
71 - idna=3.10=pyhd8ed1ab_1
72 - imagecodecs=2024.9.22=py311h58741bd_0
73 - imageio=2.37.0=pyhfb79c49_0
74 - imath=3.1.12=hbb528cf_0
75 - importlib-metadata=8.7.0=pyhe01879c_1
76 - intel-openmp=2024.2.1=h57928b3_1083
77 - ipykernel=6.30.1=py311haa95532_0
78 - ipython=9.1.0=py311haa95532_0
79 - ipython-pygments-lexers=1.1.1=py311haa95532_0
80 - ipywidgets=8.1.5=py311haa95532_0
81 - jasper=4.2.8=h8ad263b_0
82 - jedi=0.19.2=py311haa95532_0
83 - jinja2=3.1.6=py311haa95532_0
84 - joblib=1.5.2=pyhd8ed1ab_0
85 - json5=0.9.25=py311haa95532_0
86 - jsonschema=4.25.0=py311haa95532_0
87 - jsonschema-specifications=2023.7.1=py311haa95532_0
88 - jupyter=1.1.1=py311haa95532_0
89 - jupyter-lsp=2.2.5=py311haa95532_0
90 - jupyter-client=8.6.3=py311haa95532_0
91 - jupyter-console=6.6.3=py311haa95532_0
92 - jupyter_core=5.8.1=py311haa95532_0
93 - jupyter_events=0.12.0=py311haa95532_0
94 - jupyter_server=2.16.0=py311haa95532_0
95 - jupyter_server_terminals=0.5.3=py311haa95532_0
96 - jupyterlab=4.4.6=py311haa95532_0
97 - jupyterlab_pygments=0.3.0=py311haa95532_0
98 - jupyterlab_server=2.27.3=py311haa95532_0
99 - jupyterlab_widgets=3.0.15=py311haa95532_0
100 - jxrllib=1.1=hcfcfb64_3
101 - khronos-opencl-icd-loader=2024.10.24=h2466b09_1
102 - kiwisolver=1.4.9=py311h275cad7_1
103 - krb5=1.21.3=hdf4eb48_0
104 - lame=3.100=hcfcfb64_1003
105 - lazy-loader=0.4=pyhd8ed1ab_2
106 - lazy_loader=0.4=pyhd8ed1ab_2
107 - lcms2=2.17=hbcf6048_0
108 - lerc=4.0.0=h6470a55_1
109 - libabseil=20240116.2=cxx17_he0c23c2_1
110 - libaec=1.1.4=h20038f6_0
111 - libasprintf=0.22.5=h5728263_3
112 - libavif16=1.1.1=h4d049a7_2
113 - libblas=3.9.0=26_win64_mkl
114 - libbrotlicommon=1.1.0=hfd05255_4
115 - libbrotlidec=1.1.0=hfd05255_4
116 - libbrotlienc=1.1.0=hfd05255_4
117 - libcbblas=3.9.0=26_win64_mkl

```

```

118 - libclang13=21.1.1=default_ha2db4b5_0
119 - libcurl=8.14.1=h88aaa65_0
120 - libdeflate=1.22=h2466b09_0
121 - libexpat=2.7.1=hac47afa_0
122 - libffi=3.4.4=hd77b12b_1
123 - libflac=1.4.3=h63175ca_0
124 - libfreetype=2.14.0=h57928b3_1
125 - libfreetype6=2.14.0=hdbac1cb_1
126 - libgcc=15.1.0=h1383e82_5
127 - libgettextpo=0.22.5=h5728263_3
128 - libglib=2.80.2=h0df6a38_0
129 - libgomp=15.1.0=h1383e82_5
130 - libhwloc=2.12.1=default_h88281d1_1000
131 - libhwy=1.3.0=h47aaa27_0
132 - libiconv=1.18=hc1393d2_2
133 - libintl=0.22.5=h5728263_3
134 - libintl-devel=0.22.5=h5728263_3
135 - libjpeg-turbo=3.1.0=h2466b09_0
136 - libjxl=0.11.1=hb7713f0_4
137 - liblapack=3.9.0=26_win64_mkl
138 - liblapacke=3.9.0=26_win64_mkl
139 - libogg=1.3.5=h2466b09_1
140 - libopenvino=2024.2.0=hfe1841e_1
141 - libopenvino-auto-batch-plugin=2024.2.0=h04f32e0_1
142 - libopenvino-auto-plugin=2024.2.0=h04f32e0_1
143 - libopenvino-hetero-plugin=2024.2.0=h372dad0_1
144 - libopenvino-intel-cpu-plugin=2024.2.0=hfe1841e_1
145 - libopenvino-intel-gpu-plugin=2024.2.0=hfe1841e_1
146 - libopenvino-ir-frontend=2024.2.0=h372dad0_1
147 - libopenvino-onnx-frontend=2024.2.0=hdeef14f_1
148 - libopenvino-paddle-frontend=2024.2.0=hdeef14f_1
149 - libopenvino-pytorch-frontend=2024.2.0=he0c23c2_1
150 - libopenvino-tensorflow-frontend=2024.2.0=h7c40eac_1
151 - libopenvino-tensorflow-lite-frontend=2024.2.0=he0c23c2_1
152 - libopus=1.5.2=h2466b09_0
153 - libpng=1.6.50=h7351971_1
154 - libpq=12.15=h906ac69_0
155 - libprotobuf=4.25.3=h47a098d_1
156 - librosa=0.11.0=pyhd8ed1ab_0
157 - librsvg=2.56.3=hfeed636_2
158 - libsndfile=1.2.2=h81429f1_1
159 - libsodium=1.0.20=h83e8143_0
160 - libsqlite=3.50.4=hf5d6505_0
161 - libssh2=1.11.1=h9aa295b_0
162 - libtheora=1.1.1=h8cc25b3_3
163 - libtiff=4.7.0=hfc51747_1
164 - libvorbis=1.3.7=h5112557_2
165 - libwebp-base=1.6.0=h4d5522a_0
166 - libwinpthread=12.0.0.r4.gg4f2fc60ca=h57928b3_9
167 - libxcb=1.17.0=h0e4246c_0
168 - libxml2=2.13.8=h741aa76_1
169 - libxslt=1.1.43=h25c3957_0
170 - libzlib=1.3.1=h02ab6af_0
171 - libzopfli=1.0.3=h0e60522_0
172 - llvm-openmp=20.1.8=h29ce207_0
173 - llvmlite=0.44.0=py311h7c248df_2
174 - lz4-c=1.9.4=hcfcfb64_0
175 - markupsafe=3.0.2=py311h827c3e9_0
176 - matplotlib=3.10.6=py311haa95532_0
177 - matplotlib-base=3.10.6=py311h1675fdf_1
178 - matplotlib-inline=0.1.6=py311haa95532_0
179 - mistune=3.1.2=py311haa95532_0

```

```

180 - mkl=2024.2.2=h66d3029_15
181 - mkl-service=2.5.2=py311h21a4f29_0
182 - mkl_fft=2.0.0=py311h1dea9b0_0
183 - mkl_random=1.2.10=py311h4dd216f_0
184 - mpg123=1.32.9=h01009b0_0
185 - msgpack-python=1.1.1=py311h3fd045d_1
186 - munkres=1.1.4=pyhd8ed1ab_1
187 - nbclient=0.10.2=py311haa95532_0
188 - nbconvert=7.16.6=py311haa95532_0
189 - nbconvert-core=7.16.6=py311haa95532_0
190 - nbconvert-pandoc=7.16.6=py311haa95532_0
191 - nbformat=5.10.4=py311haa95532_0
192 - nest-asyncio=1.6.0=py311haa95532_0
193 - networkx=3.5=pyhe01879c_0
194 - notebook=7.4.5=py311haa95532_0
195 - notebook-shim=0.2.4=py311haa95532_0
196 - numba=0.61.2=py311h7afb941_1
197 - numexpr=2.12.1=mkl_py311hcc47716_0
198 - numpy=2.2.6=py311h5e411d1_0
199 - opencl-headers=2025.06.13=he0c23c2_0
200 - openexr=3.2.2=h9aba623_2
201 - openh264=2.5.0=ha9db3cd_0
202 - openjpeg=2.5.3=h24db6dd_1
203 - openssl=3.5.2=h725018a_0
204 - overrides=7.4.0=py311haa95532_0
205 - packaging=25.0=pyh29332c3_1
206 - pandas=2.3.2=py311ha5e6156_0
207 - pandoc=2.12=haa95532_3
208 - pandocfilters=1.5.1=py311haa95532_0
209 - pango=1.50.7=haaec5b9_2
210 - parso=0.8.4=py311haa95532_0
211 - pcre2=10.43=h17e33f8_0
212 - pillow=11.3.0=py311h0f9b5fc_1
213 - pip=25.2=pyhc872135_0
214 - pixman=0.46.4=h5112557_1
215 - platformdirs=4.4.0=pyhcf101f3_0
216 - pooch=1.8.2=pyhd8ed1ab_3
217 - prometheus-client=0.21.1=py311haa95532_0
218 - prompt-toolkit=3.0.43=py311haa95532_0
219 - prompt_toolkit=3.0.43=hd3eb1b0_0
220 - psutil=7.0.0=py311h02ab6af_0
221 - pthread-stubs=0.4=h0e40799_1002
222 - pugixml=1.14=h63175ca_0
223 - pure_eval=0.2.3=py311haa95532_0
224 - pycparser=2.22=pyh29332c3_1
225 - pygments=2.19.1=py311haa95532_0
226 - pyparsing=3.2.3=pyhe01879c_2
227 - pyqt=6.7.1=py311h5da7b33_0
228 - pyqt6-sip=13.9.1=py311h827c3e9_0
229 - pysocks=1.7.1=pyh09c184e_7
230 - pysoundfile=0.13.1=pyhd8ed1ab_0
231 - python=3.11.13=h981015d_0
232 - python-dateutil=2.9.0.post0=pyhe01879c_2
233 - python-fastjsonschema=2.20.0=py311haa95532_0
234 - python-json-logger=3.2.1=py311haa95532_0
235 - python-tzdata=2025.2=pyhd3eb1b0_0
236 - python_abi=3.11=2_cp311
237 - pytz=2025.2=py311haa95532_0
238 - pywavelets=1.9.0=py311h17033d2_1
239 - pywin32=311=py311h885b0b7_0
240 - pywinpty=2.0.15=py311h72d21ff_0
241 - pyyaml=6.0.2=py311h827c3e9_0

```



```

242 - pyzmq=26.2.0=py311h5da7b33_0
243 - qhull=2020.2=hc790b64_5
244 - qt6-main=6.7.2=h913a85e_3
245 - qtbase=6.7.2=h0804d20_0
246 - qtconsole=5.6.1=py311haa95532_1
247 - qtdeclarative=6.7.2=h5da7b33_0
248 - qtpy=2.4.3=py311haa95532_0
249 - qtsvg=6.7.2=hf2fb9eb_0
250 - qttools=6.7.2=h0de5f00_0
251 - qtwebchannel=6.7.2=h5da7b33_0
252 - qtwebsockets=6.7.2=h5da7b33_0
253 - rav1e=0.6.6=h975169c_2
254 - referencing=0.30.2=py311haa95532_0
255 - requests=2.32.5=pyhd8ed1ab_0
256 - rfc3339-validator=0.1.4=py311haa95532_0
257 - rfc3986-validator=0.1.1=py311haa95532_0
258 - rpds-py=0.22.3=py311h636fa0f_0
259 - scikit-image=0.25.2=py311h11fd7f3_2
260 - scikit-learn=1.7.2=py311h8a15ebc_0
261 - scipy=1.16.2=py311h9a1c30b_0
262 - send2trash=1.8.2=py311haa95532_1
263 - setuptools=72.1.0=py311haa95532_0
264 - sip=6.10.0=py311h5da7b33_0
265 - six=1.17.0=pyhe01879c_1
266 - snappy=1.2.2=h7fa0ca8_0
267 - sniffio=1.3.0=py311haa95532_0
268 - soupsieve=2.5=py311haa95532_0
269 - soxr=0.1.3=hcfcfb64_3
270 - soxr-python=1.0.0=py311h3e6a449_0
271 - sqlite=3.50.2=hda9a48d_1
272 - stack_data=0.6.3=py311haa95532_0
273 - standard-aifc=3.13.0=py311h1ea47a8_2
274 - standard-sunau=3.13.0=py311h1ea47a8_2
275 - svt-av1=2.3.0=he0c23c2_0
276 - tbb=2021.13.0=h18a62a1_3
277 - tbb-devel=2021.13.0=h4eb897c_3
278 - terminado=0.17.1=py311haa95532_0
279 - threadpoolctl=3.6.0=pyhecae5ae_0
280 - tifffile=2024.12.12=pyhd8ed1ab_0
281 - tinycss2=1.4.0=py311haa95532_0
282 - tk=8.6.15=hf199647_0
283 - tornado=6.5.2=py311h3485c13_1
284 - traitlets=5.14.3=py311haa95532_0
285 - typing-extensions=4.15.0=py311haa95532_0
286 - typing_extensions=4.15.0=py311haa95532_0
287 - tzdata=2025b=h04d1e81_0
288 - ucrt=10.0.22621.0=haa95532_0
289 - unicodedata2=16.0.0=py311h3485c13_1
290 - urllib3=2.5.0=pyhd8ed1ab_0
291 - vc=14.42=haa95532_5
292 - vc14_runtime=14.44.35208=h4927774_10
293 - vs2015_runtime=14.44.35208=ha6b5a95_10
294 - wcwidth=0.2.13=py311haa95532_0
295 - webencodings=0.5.1=py311haa95532_1
296 - websocket-client=1.8.0=py311haa95532_0
297 - wheel=0.45.1=py311haa95532_0
298 - widgetsnbextension=4.0.13=py311haa95532_0
299 - win_inet_pton=1.1.0=pyh7428d3b_8
300 - winpty=0.4.3=4
301 - x264=1!164.3095=h8ffe710_2
302 - x265=3.5=h2d74725_3
303 - xorg-libxau=1.0.12=h0e40799_0

```

```
304 - xorg-libxdmcp=1.1.5=h0e40799_0
305 - xz=5.6.4=h4754444_1
306 - yaml=0.2.5=he774522_0
307 - zeromq=4.3.5=h6c54ac7_1
308 - zfp=1.0.1=h2f0f97f_3
309 - zipp=3.23.0=pyhd8ed1ab_0
310 - zlib=1.3.1=h02ab6af_0
311 - zlib-ng=2.2.5=h1608b31_0
312 - zstandard=0.24.0=py311h2d646e2_1
313 - zstd=1.5.7=hbeecb71_2
314 - pip:
315     - imageio-ffmpeg==0.6.0
316     - moviepy==2.2.1
317     - opencv-python==4.12.0.88
318     - opencv-python-headless==4.12.0.88
319     - proglog==0.1.12
320     - python-dotenv==1.1.1
321     - tqdm==4.67.1
322 prefix: C:\Anaconda3\envs\multimedia
323 ]
```

Kode 12: 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

[Pertamkali setup python environment untuk multimedia tidak terlalu sulit, walaupun sedikit terkendala tapi masih bisa di perbaiki. Setelah melakukan setup ini saya jadi banyak belajar, saya pikir multimedia sebatas gambar, audio, dan video saja ternyata berawal dari itu banyak hal yang bisa diciptakan. Untuk projek kedepannya mungkin akan semakin terbiasa dalam menggunakan library multimedia didalam environment yang sudah dibuat. Saran untuk teman - teman, ada baiknya walaupun mengerjakan tugas dibantu oleh LLM sebaiknya baca juga hasil output dari kode yang dicoba, jangan next - next saja, belajar untuk teliti.]

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

References

[Tautan percakapan ChatGPT](#)