



Nama: **Zefanya Danovanta Tarigan (122140101)** 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: **Menggunakan Conda**

- Screenshot atau copy-paste output dari perintah verifikasi environment

```

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

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

Loading personal and system profiles took 4546ms.
(base) PS C:\WINDOWS\system32> conda create -n multimedia python=3.11
3 channel Terms of Service accepted
WARNING: A conda environment already exists at 'C:\Users\Lenovo\miniconda3\envs\multimedia'

Remove existing environment?
This will remove ALL directories contained within this specified prefix directory, including any other conda environments.

(y/[n])? y

Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\Lenovo\miniconda3\envs\multimedia

added / updated specs:
- python=3.11

The following NEW packages will be INSTALLED:

bz2      pkgs/main/win-64::bz2-1.0.8-h2b6ff1b_6
ca-certificates pkgs/main/win-64::ca-certificates-2025.7.15-haa95532_0
expat    pkgs/main/win-64::expat-2.7.1-h8ddb27b_0
libffi   pkgs/main/win-64::libffi-3.4.4-hd77b12b_1
openssl  pkgs/main/win-64::openssl-3.0.17-h35632f6_0
pip      pkgs/main/noarch::pip-25.1-pyh872135_2
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-hda9a48d_1

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-1.2.13-h8cc25b3_1

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) PS C:\WINDOWS\system32> conda activate multimedia
(multimedia) PS C:\WINDOWS\system32> conda info --envs

# conda environments:
#
base             C:\Users\Lenovo\miniconda3
multimedia        * C:\Users\Lenovo\miniconda3\envs\multimedia

(multimedia) PS C:\WINDOWS\system32>

```

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 : conda install -c conda-forge librosa pysoundfile scipy opencv scikit-image ffmpeg pandas jupyter matplotlib pillow
 - Screenshot proses instalasi atau output sukses

```
(multimedia) PS C:\WINDOWS\system32> conda install -c conda-forge librosa pysoundfile scipy opencv pillow scikit-image matplotlib ffmpeg numpy pandas jupyter
>>
1 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:\Users\Lenovo\miniconda3\envs\multimedia

added / updated specs:
- ffmpeg
- jupyter
- librosa
- matplotlib
- numpy
- opencv
- pandas
- pillow
- pysoundfile
- scikit-image
- scipy

The following packages will be downloaded:
```

package	build	size	source
_libvifc_api-1.3.0	h57928b3_2	10 KB	conda-forge
anyio-4.10.0	pyha01579c_0	132 KB	conda-forge
asn-3.9.1	h80c23c2_0	1.9 MB	conda-forge
argon2-cffi-25.1.0	pyhd8ed1ab_0	18 KB	conda-forge
argon2-cffi-bindings-25.1.0	py311h3485c13_0	38 KB	conda-forge
arrow-1.3.0	pyhd8ed1ab_1	98 KB	conda-forge
asttokens-3.0.0	pyhd8ed1ab_1	28 KB	conda-forge
async-lru-2.0.5	pyh29332c3_0	17 KB	conda-forge

- Daftar library yang berhasil diinstall dengan versinya

1. librosa: 0.11.0
2. soundfile: 0.13.1
3. scipy: 1.15.3
4. opencv-python: 4.12.0
5. scikit-image: 0.25.2
6. matplotlib: 3.10.5
7. moviepy: 2.2.1
8. numpy: 1.22.4
9. pandas: 2.3.2
10. jupyter: 1.1.1

3.4 Bagian 3: Verifikasi Instalasi

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

```

3  def verify_libraries():
4      """
5      Fungsi untuk memeriksa jika
6      """
7      # Daftar library yang ingin
8      required_libraries = [
9          'librosa',
10         'soundfile',
11         'scipy',
12         'cv2',
13         'skinage',
14         'matplotlib',
15         'moviepy',
16         'jupyter',
17         'numpy',
18         'pandas'

```

PROBLEMS OUTPUT DEBUG CONSOLE TEST

Versi Python yang digunakan: 3.10.16

```

[ ✓ ] librosa      : Terinstall (0.
[ ✓ ] soundfile   : Terinstall (0.
[ ✓ ] scipy       : Terinstall (1.
[ ✓ ] cv2         : Terinstall (4.
[ ✓ ] skinage     : Terinstall (0.
[ ✓ ] matplotlib : Terinstall (3.
[ ✓ ] moviepy     : Terinstall (2.
[ ✓ ] jupyter     : Terinstall ( )
[ ✓ ] numpy       : Terinstall (2.
[ ✓ ] pandas      : Terinstall (2.

```

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}")

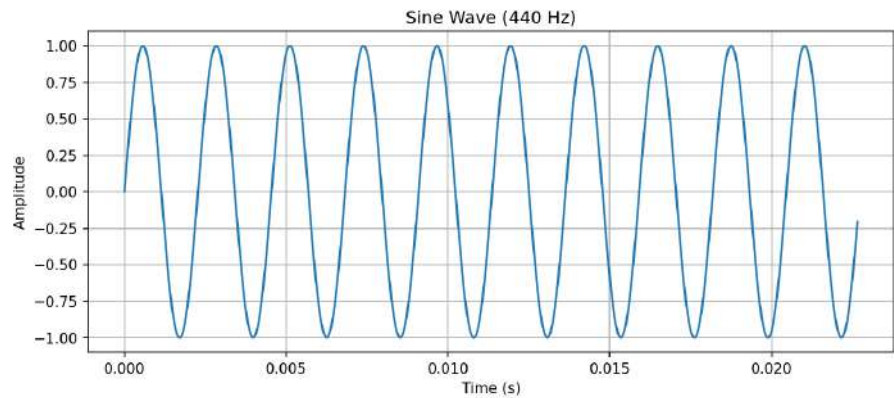
```

Kode 9: Test image processing sederhana

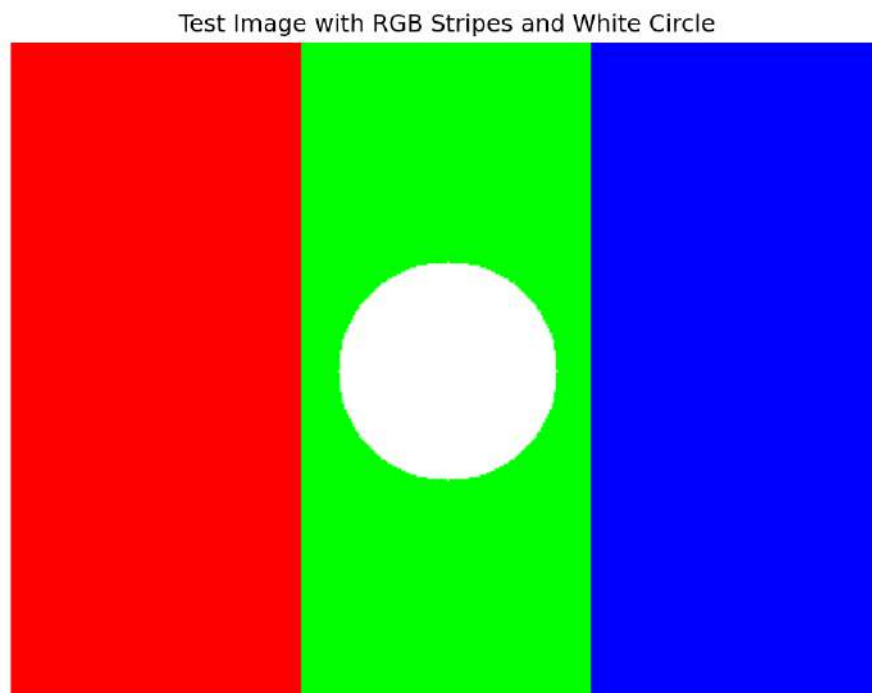
Dokumentasikan hasil eksekusi:

- Screenshot output dari kedua script di atas

```
(multimedia) PS C:\Users\Lenovo\Downloads\Worksheet 1\Worksheet 1> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python38/python.exe "c:/Users/Lenovo/Downloads/Worksheet 1/Worksheet 1/test_multimedia.py"
Created test image: 400x300 pixels
Image shape: (300, 400, 3)
Image dtype: uint8
(multimedia) PS C:\Users\Lenovo\Downloads\Worksheet 1\Worksheet 1> █
```



- Gambar yang dihasilkan



4 Bagian Laporan

4.1 Output Verifikasi Instalasi

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

```
1 Audio Signal :
2 Generated 2s sine wave at 440Hz
3 Sample rate: 44100Hz
4 Total samples: 88200
5
6 Image :
7 Created test image: 400x300 pixels
8 Image shape: (300, 400, 3)
```


9 Image dtype: uint8

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?

Membuat lingkungan terpisah itu penting untuk menghindari konflik antara dependensi yang diperlukan oleh berbagai proyek multimedia. memisahkan lingkungan, bisa memastikan bahwa setiap proyek menggunakan paket dan versi yang sesuai tanpa saling mengganggu.

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

- **Conda** : Memiliki ukuran besar karena distribusi Anaconda yang mencakup banyak paket
- **Venv** : built-in Python, ringan, tapi perlu install library manual.
- **UV** : Paling Cepat

3. Library mana yang paling sulit diinstall dan mengapa?

Tidak Ada library yang sulit diinstall

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

Akan Memperbarui Ke versi yang terbaru atau menyesuaikan dengan versi library yang sesuai

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

- Matplotlib : Untuk visualisasi gambar berbentuk grafik
- Numpy : untuk komputasi numerik matematika
- Pandas : untuk membaca dan memanipulasi dataset
- OpenCV : untuk pengolahan citra
- Scikit-Image : untuk algoritma pengolahan citra
- Jupyter : untuk membuat dan berbagi dokumen
- Soundfile : untuk membaca dan menulis file audio
- Librosa : untuk menganalisis audio
- Scipy : untuk pengelolaan fungsi algoritma matematika yang lebih rumit
- Moviepy : untuk mengedit dan memproses file video

4.4 Troubleshooting

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

- -

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 [name: multimedia
2 channels:
3   - conda-forge
4   - defaults
5 dependencies:
6   - _libavif_api=1.3.0=h57928b3_2
7   - anyio=4.10.0=pyhe01879c_0
8   - aom=3.9.1=he0c23c2_0
9   - argon2-cffi=25.1.0=pyhd8ed1ab_0
10  - argon2-cffi-bindings=25.1.0=py311h3485c13_0
11  - arrow=1.3.0=pyhd8ed1ab_1
12  - asttokens=3.0.0=pyhd8ed1ab_1
13  - async-lru=2.0.5=pyh29332c3_0
14  - attrs=25.3.0=pyh71513ae_0
15  - audioread=3.0.1=py311h1ea47a8_2
16  - babel=2.17.0=pyhd8ed1ab_0
17  - beautifulsoup4=4.13.5=pyha770c72_0
18  - bleach=6.2.0=pyh29332c3_4
19  - bleach-with-css=6.2.0=h82add2a_4
20  - blosc=1.21.6=h4190f5b_0
21  - brotli=1.0.9=hcfcfb64_9
22  - brotli-bin=1.0.9=hcfcfb64_9
23  - brotli-python=1.0.9=py311h12c1d0e_9
24  - bzip2=1.0.8=h2bbff1b_6
25  - ca-certificates=2025.8.3=h4c7d964_0
26  - cached-property=1.5.2=hd8ed1ab_1
27  - cached_property=1.5.2=pyha770c72_1
28  - cairo=1.18.4=he9e932c_0
29  - certifi=2025.8.3=pyhd8ed1ab_0
30  - cffi=1.17.1=py311he736701_0
31  - cfitsio=3.470=h0af3d06_7
32  - charls=2.2.0=h39d44d4_0
33  - charset-normalizer=3.4.3=pyhd8ed1ab_0
34  - colorama=0.4.6=pyhd8ed1ab_1
35  - comm=0.2.3=pyhe01879c_0
36  - contourpy=1.3.3=py311h3fd045d_1
```

```

37 - cpython=3.11.13=py311hd8ed1ab_0
38 - cycpler=0.12.1=pyhd8ed1ab_1
39 - dav1d=1.2.1=hcfcfb64_0
40 - debugpy=1.8.16=py311h5dfdf8_0
41 - decorator=5.2.1=pyhd8ed1ab_0
42 - defusedxml=0.7.1=pyhd8ed1ab_0
43 - eigen=3.4.0=h91493d7_0
44 - exceptiongroup=1.3.0=pyhd8ed1ab_0
45 - executing=2.2.0=pyhd8ed1ab_0
46 - expat=2.7.1=h8ddb27b_0
47 - fontconfig=2.14.1=hb33846d_3
48 - fonttools=4.59.2=py311h3f79411_0
49 - fqdn=1.5.1=pyhd8ed1ab_1
50 - freeglut=3.4.0=h8a1e904_1
51 - freetype=2.13.3=h0620614_0
52 - fribid=1.0.10=h8d14728_0
53 - gflags=2.2.2=he0c23c2_1005
54 - giflib=5.2.2=h64bf75a_0
55 - glog=0.5.0=h4797de2_0
56 - graphite2=1.3.14=hac47afa_2
57 - gst-plugins-base=1.24.12=h91a6125_1
58 - gstreamer=1.24.12=hfb93a4f_1
59 - gstreamer-orc=0.4.41=h1f81b68_0
60 - h11=0.16.0=pyhd8ed1ab_0
61 - h2=4.3.0=pyhcf101f3_0
62 - harfbuzz=10.2.0=he2f9f60_1
63 - hdf5=1.14.5=ha36df97_2
64 - hpack=4.1.0=pyhd8ed1ab_0
65 - httpcore=1.0.9=pyh29332c3_0
66 - httpx=0.28.1=pyhd8ed1ab_0
67 - hyperframe=6.1.0=pyhd8ed1ab_0
68 - icc_rt=2022.1.0=h6049295_2
69 - icu=73.2=h63175ca_0
70 - idna=3.10=pyhd8ed1ab_1
71 - imagecodecs=2024.9.22=py311hf5f7160_2
72 - imageio=2.37.0=pyhfb79c49_0
73 - importlib-metadata=8.7.0=pyhe01879c_1
74 - ipykernel=6.30.1=pyh3521513_0
75 - ipython=9.5.0=pyh6be1c34_0
76 - ipython-pygments-lexers=1.1.1=pyhd8ed1ab_0
77 - ipywidgets=8.1.7=pyhd8ed1ab_0
78 - isoduration=20.11.0=pyhd8ed1ab_1
79 - jedi=0.19.2=pyhd8ed1ab_1
80 - jinja2=3.1.6=pyhd8ed1ab_0
81 - joblib=1.5.2=pyhd8ed1ab_0
82 - jpeg=9e=hcfcfb64_3
83 - json5=0.12.1=pyhd8ed1ab_0
84 - jsonpointer=3.0.0=py311h1ea47a8_1
85 - jsonschema=4.25.1=pyhe01879c_0
86 - jsonschema-specifications=2025.4.1=pyh29332c3_0
87 - jsonschema-with-format-nongpl=4.25.1=he01879c_0
88 - jupyter=1.1.1=pyhd8ed1ab_1
89 - jupyter-lsp=2.3.0=pyhcf101f3_0
90 - jupyter-client=8.6.3=pyhd8ed1ab_1
91 - jupyter-console=6.6.3=pyhd8ed1ab_1
92 - jupyter-core=5.8.1=pyh5737063_0
93 - jupyter-events=0.12.0=pyh29332c3_0
94 - jupyter-server=2.17.0=pyhcf101f3_0
95 - jupyter-server-terminals=0.5.3=pyhd8ed1ab_1
96 - jupyterlab=4.4.6=pyhd8ed1ab_0
97 - jupyterlab-pygments=0.3.0=pyhd8ed1ab_2
98 - jupyterlab-server=2.27.3=pyhd8ed1ab_1

```

```

99 - jupyterlab_widgets=3.0.15=pyhd8ed1ab_0
100 - kiwisolver=1.4.9=py311h275cad7_1
101 - krb5=1.21.3=hdf4eb48_0
102 - lame=3.100=hcfcfb64_1003
103 - lark=1.2.2=pyhd8ed1ab_1
104 - lazy-loader=0.4=pyhd8ed1ab_2
105 - lazy_loader=0.4=pyhd8ed1ab_2
106 - lcms2=2.16=h62be587_1
107 - lerc=4.0.0=h6470a55_1
108 - libabseil=20250127.0=cxx17_h4eb7d71_0
109 - libaec=1.1.4=h20038f6_0
110 - libavif=1.3.0=he916da2_2
111 - libavif16=1.3.0=he916da2_2
112 - libblas=3.9.0=34_h5709861_mkl
113 - libbrotlicommon=1.0.9=hcfcfb64_9
114 - libbrotlidec=1.0.9=hcfcfb64_9
115 - libbrotlienc=1.0.9=hcfcfb64_9
116 - libcbblas=3.9.0=34_h2a3cdd5_mkl
117 - libclang13=14.0.6=default_h8e68704_2
118 - libdeflate=1.22=h2466b09_0
119 - libffi=3.4.4=hd77b12b_1
120 - libflac=1.4.3=h63175ca_0
121 - libglib=2.84.2=h405b238_0
122 - libhwloc=2.12.1=default_h88281d1_1000
123 - libiconv=1.18=hc1393d2_2
124 - libkrb5=1.21.3=h885b0b7_4
125 - liblapack=3.9.0=34_hf9ab0e9_mkl
126 - libogg=1.3.5=h2466b09_1
127 - libopus=1.5.2=h2466b09_0
128 - libpng=1.6.39=h8cc25b3_0
129 - libpq=17.4=h4a159e6_2
130 - libprotobuf=5.29.3=h65a231f_1
131 - librosa=0.11.0=pyhd8ed1ab_0
132 - libsndfile=1.2.2=h81429f1_1
133 - libsodium=1.0.20=hc70643c_0
134 - libtheora=1.2.0=habf309d_0
135 - libtiff=4.7.0=h404307b_0
136 - libvorbis=1.3.7=h5112557_2
137 - libwebp-base=1.6.0=h4d5522a_0
138 - libwinpthread=12.0.0.r4.gg4f2fc60ca=h57928b3_9
139 - libxml2=2.13.8=h866ff63_0
140 - libxslt=1.1.43=h25c3957_0
141 - libzopfli=1.0.3=h0e60522_0
142 - llvm-openmp=20.1.8=hfa2b4ca_2
143 - llvmlite=0.44.0=py311h8b1c7eb_1
144 - lz4-c=1.9.4=hcfcfb64_0
145 - markupsafe=3.0.2=py311h5082efb_1
146 - matplotlib=3.10.5=py311h1ea47a8_0
147 - matplotlib-base=3.10.5=py311h43afe63_0
148 - matplotlib-inline=0.1.7=pyhd8ed1ab_1
149 - minizip=4.0.3=hb68bac4_0
150 - mistune=3.1.4=pyhcf101f3_0
151 - mkl=2024.2.2=h57928b3_16
152 - mpg123=1.32.9=h01009b0_0
153 - msgpack-python=1.1.1=py311h3257749_0
154 - munkres=1.1.4=pyhd8ed1ab_1
155 - nbclient=0.10.2=pyhd8ed1ab_0
156 - nbconvert-core=7.16.6=pyh29332c3_0
157 - nbformat=5.10.4=pyhd8ed1ab_1
158 - nest-asyncio=1.6.0=pyhd8ed1ab_1
159 - networkx=3.5=pyhe01879c_0
160 - notebook=7.4.5=pyhd8ed1ab_0

```

```

161 - notebook-shim=0.2.4=pyhd8ed1ab_1
162 - numba=0.61.2=py311h7afb941_1
163 - numpy=2.2.6=py311h5e411d1_0
164 - opencv=4.10.0=py311h28596fa_7
165 - openjpeg=2.5.2=h9b5d1b5_1
166 - openssl=3.5.2=h725018a_0
167 - overrides=7.7.0=pyhd8ed1ab_1
168 - packaging=25.0=pyh29332c3_1
169 - pandas=2.3.2=py311h11fd7f3_0
170 - pandocfilters=1.5.0=pyhd8ed1ab_0
171 - parso=0.8.5=pyhcf101f3_0
172 - pcre2=10.42=h0ff8eda_1
173 - pickleshare=0.7.5=pyhd8ed1ab_1004
174 - pillow=11.3.0=py311hb328d1f_0
175 - pip=25.1=pyhc872135_2
176 - pixman=0.46.4=h5112557_1
177 - platformdirs=4.4.0=pyhcf101f3_0
178 - pooch=1.8.2=pyhd8ed1ab_3
179 - prometheus_client=0.22.1=pyhd8ed1ab_0
180 - prompt-toolkit=3.0.52=pyha770c72_0
181 - prompt_toolkit=3.0.52=hd8ed1ab_0
182 - psutil=7.0.0=py311h3485c13_1
183 - pure_eval=0.2.3=pyhd8ed1ab_1
184 - pycparser=2.22=pyh29332c3_1
185 - pygments=2.19.2=pyhd8ed1ab_0
186 - pyparsing=3.2.3=pyhe01879c_2
187 - pyside6=6.7.3=py311h28b127d_1
188 - pysocks=1.7.1=pyh09c184e_7
189 - python=3.11.13=h981015d_0
190 - python-dateutil=2.9.0.post0=pyhe01879c_2
191 - python-fastjsonschema=2.21.2=pyhe01879c_0
192 - python-json-logger=2.0.7=pyhd8ed1ab_0
193 - python-tzdata=2025.2=pyhd8ed1ab_0
194 - python_abi=3.11=2_cp311
195 - pytz=2025.2=pyhd8ed1ab_0
196 - pywavelets=1.9.0=py311h17033d2_1
197 - pywin32=311=py311hefeebc8_1
198 - pywinpty=2.0.15=py311hda3d55a_0
199 - pyyaml=6.0.2=py311h5082efb_2
200 - pyzmq=27.0.2=py311hb77b9c8_2
201 - qtbase=6.7.3=hd088775_4
202 - qtdeclarative=6.7.3=h885b0b7_1
203 - qtshadertools=6.7.3=h885b0b7_1
204 - qtsvg=6.7.3=h9d4b640_1
205 - qttools=6.7.3=hcb596f7_1
206 - qtwebchannel=6.7.3=h885b0b7_1
207 - qtwebengine=6.7.3=h3869032_1
208 - qtwebsockets=6.7.3=h885b0b7_1
209 - ravle=0.7.1=ha073cba_3
210 - referencing=0.36.2=pyh29332c3_0
211 - requests=2.32.5=pyhd8ed1ab_0
212 - rfc3339-validator=0.1.4=pyhd8ed1ab_1
213 - rfc3986-validator=0.1.1=pyh9f0ad1d_0
214 - rfc3987-syntax=1.1.0=pyhe01879c_1
215 - rpds-py=0.27.1=py311hf51aa87_0
216 - scikit-image=0.25.2=py311hcf9f919_0
217 - scikit-learn=1.7.1=py311h8a15ebc_0
218 - scipy=1.16.1=py311h9a1c30b_1
219 - send2trash=1.8.3=pyh5737063_1
220 - setuptools=78.1.1=py311haa95532_0
221 - six=1.17.0=pyhe01879c_1
222 - snappy=1.2.2=h7fa0ca8_0

```

```

223 - sniffio=1.3.1=pyhd8ed1ab_1
224 - soupsieve=2.8=pyhd8ed1ab_0
225 - soxr=0.1.3=hcfcfb64_3
226 - soxr-python=0.5.0.post1=py311hda3d55a_1
227 - sqlite=3.50.2=hda9a48d_1
228 - stack_data=0.6.3=pyhd8ed1ab_1
229 - standard-aifc=3.13.0=py311h1ea47a8_2
230 - standard-sunau=3.13.0=py311h1ea47a8_2
231 - svt-av1=3.1.2=hac47afa_0
232 - tbb=2021.13.0=h18a62a1_3
233 - terminado=0.18.1=pyh5737063_0
234 - threadpoolctl=3.6.0=pyhecae5ae_0
235 - tifffile=2024.12.12=pyhd8ed1ab_0
236 - tinycss2=1.4.0=pyhd8ed1ab_0
237 - tk=8.6.15=hf199647_0
238 - toml=2.2.1=pyhe01879c_2
239 - tornado=6.5.2=py311h3485c13_0
240 - traitlets=5.14.3=pyhd8ed1ab_1
241 - types-python-dateutil=2.9.0.20250822=pyhd8ed1ab_0
242 - typing-extensions=4.15.0=h396c80c_0
243 - typing_extensions=4.15.0=pyhcf101f3_0
244 - typing_utils=0.1.0=pyhd8ed1ab_1
245 - tzdata=2025b=h04d1e81_0
246 - ucrt=10.0.22621.0=haa95532_0
247 - unicodedata2=16.0.0=py311h3485c13_1
248 - uri-template=1.3.0=pyhd8ed1ab_1
249 - urllib3=2.5.0=pyhd8ed1ab_0
250 - vc=14.3=h2df5915_10
251 - vc14_runtime=14.44.35208=h4927774_10
252 - vs2015_runtime=14.44.35208=ha6b5a95_10
253 - wcwidth=0.2.13=pyhd8ed1ab_1
254 - webcolors=24.11.1=pyhd8ed1ab_0
255 - webencodings=0.5.1=pyhd8ed1ab_3
256 - websocket-client=1.8.0=pyhd8ed1ab_1
257 - wheel=0.45.1=py311haa95532_0
258 - widgetsnbextension=4.0.14=pyhd8ed1ab_0
259 - win_inet_pton=1.1.0=pyh7428d3b_8
260 - winpty=0.4.3=4
261 - xz=5.6.4=h4754444_1
262 - yaml=0.2.5=h6a83c73_3
263 - zeromq=4.3.5=ha9f60a1_7
264 - zfp=1.0.1=h2f0f97f_3
265 - zipp=3.23.0=pyhd8ed1ab_0
266 - zlib=1.2.13=h8cc25b3_1
267 - zlib-ng=2.0.7=hcfcfb64_0
268 - zstandard=0.23.0=py311h3485c13_3
269 - zstd=1.5.6=h8880b57_0
270 - pip:
271   - ffmpeg==1.4
272   - ffmpeg-python==0.2.0
273   - future==1.0.0
274   - pysoundfile==0.9.0.post1
275 prefix: C:\Users\Lenovo\miniconda3\envs\multimedia
276 ]

```

Kode 13: Environment/Requirements file

6 Kesimpulan

Setup python environment buat mata kuliah multimedia ini memberikan saya pengetahuan yang menarik karena pengembangan multimedia berbasis python dapat dijalankan. saya dapat mudah mengelola paket

dependency. untuk proyek multimedia selanjutnya bisa disiapkan berupa dependency yang lebih spesifik sesuai dengan kebutuhan yang tugaskan nanti

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

References

<https://chatgpt.com/share/68b2d63e-f120-800b-b652-3b0e495736fd>