Program Studi Teknik Informatika
Institut Teknologi Sumatera

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 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:

```
# Membuat environment baru dengan nama 'multimedia'
conda create -n multimedia python=3.11

# Mengaktifkan environment
conda activate multimedia

# Verifikasi environment aktif
conda info --envs
```

Kode 1: Membuat environment dengan Conda

3.2.2 Opsi 2: Menggunakan venv (Built-in Python)

```
# Membuat environment baru
python3 -m venv multimedia-env

# Mengaktifkan environment (Linux/Mac)
source multimedia-env/bin/activate

# Mengaktifkan environment (Windows)
# multimedia-env\Scripts\activate

# Verifikasi environment aktif
which python
```

Kode 2: Membuat environment dengan venv

3.2.3 Opsi 3: Menggunakan uv (Modern dan cepat)

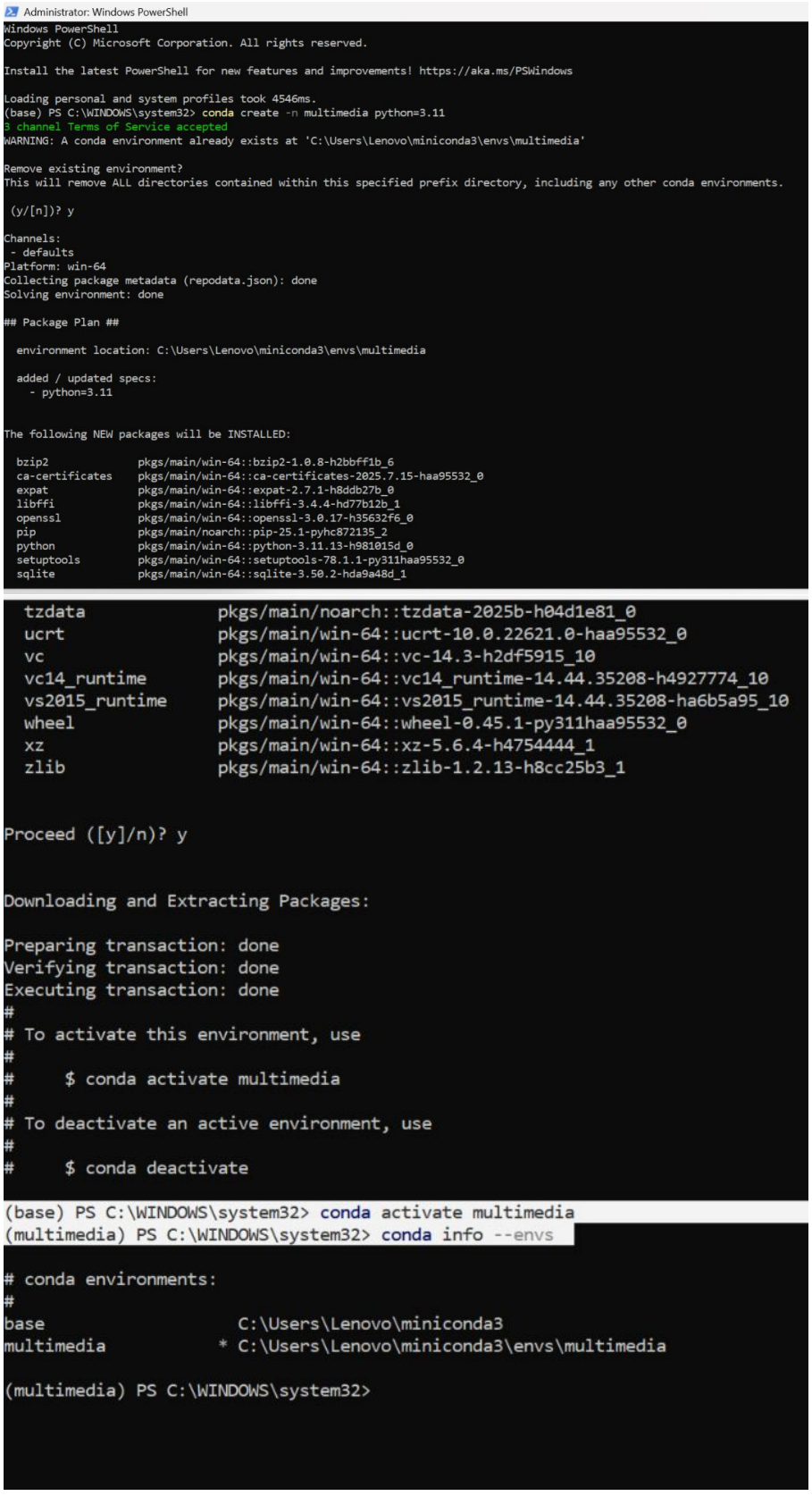
```
# Install uv terlebih dahulu jika belum ada
# pip install uv

# Membuat environment baru
uv venv multimedia-uv

# Mengaktifkan environment (Linux/Mac)
source multimedia-uv/bin/activate

# Mengaktifkan environment (Windows)
# multimedia-uv\Scripts\activate

# Verifikasi environment aktif
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



3.3 Bagian 2: Instalasi Library Multimedia

Setelah environment aktif, install library-library berikut:

3.3.1 Library Audio Processing

```
# Untuk conda:
conda install -c conda-forge librosa soundfile scipy

# Untuk pip (venv/uv):
pip install librosa soundfile scipy
```

Kode 4: Instalasi library audio

3.3.2 Library Image Processing

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

# Untuk pip (venv/uv):
pip install opencv-python pillow scikit-image matplotlib
```

Kode 5: Instalasi library image

3.3.3 Library Video Processing

```
# Untuk conda:
conda install -c conda-forge ffmpeg
pip install moviepy

# Untuk pip (venv/uv):
pip install moviepy
```

Kode 6: Instalasi library video

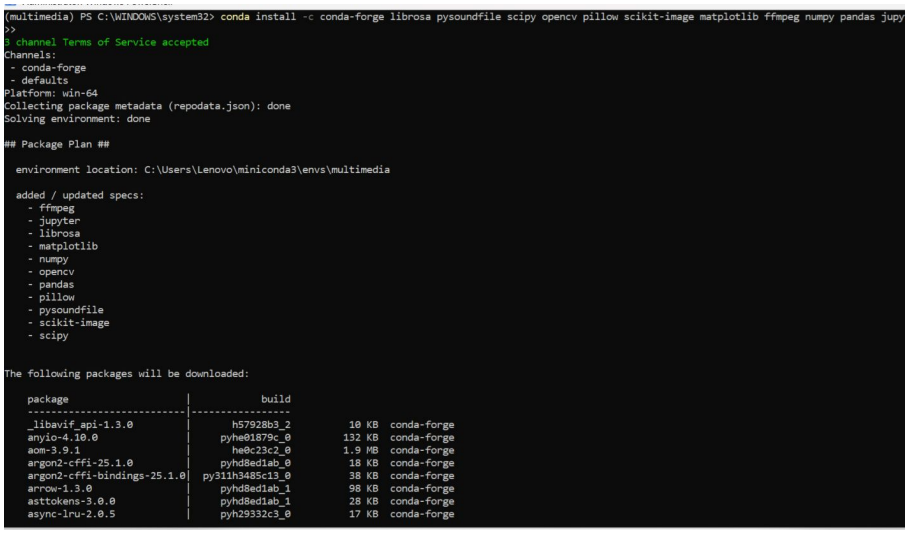
3.3.4 Library General Purpose

```
# Untuk conda:
conda install numpy pandas jupyter

# Untuk pip (venv/uv):
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



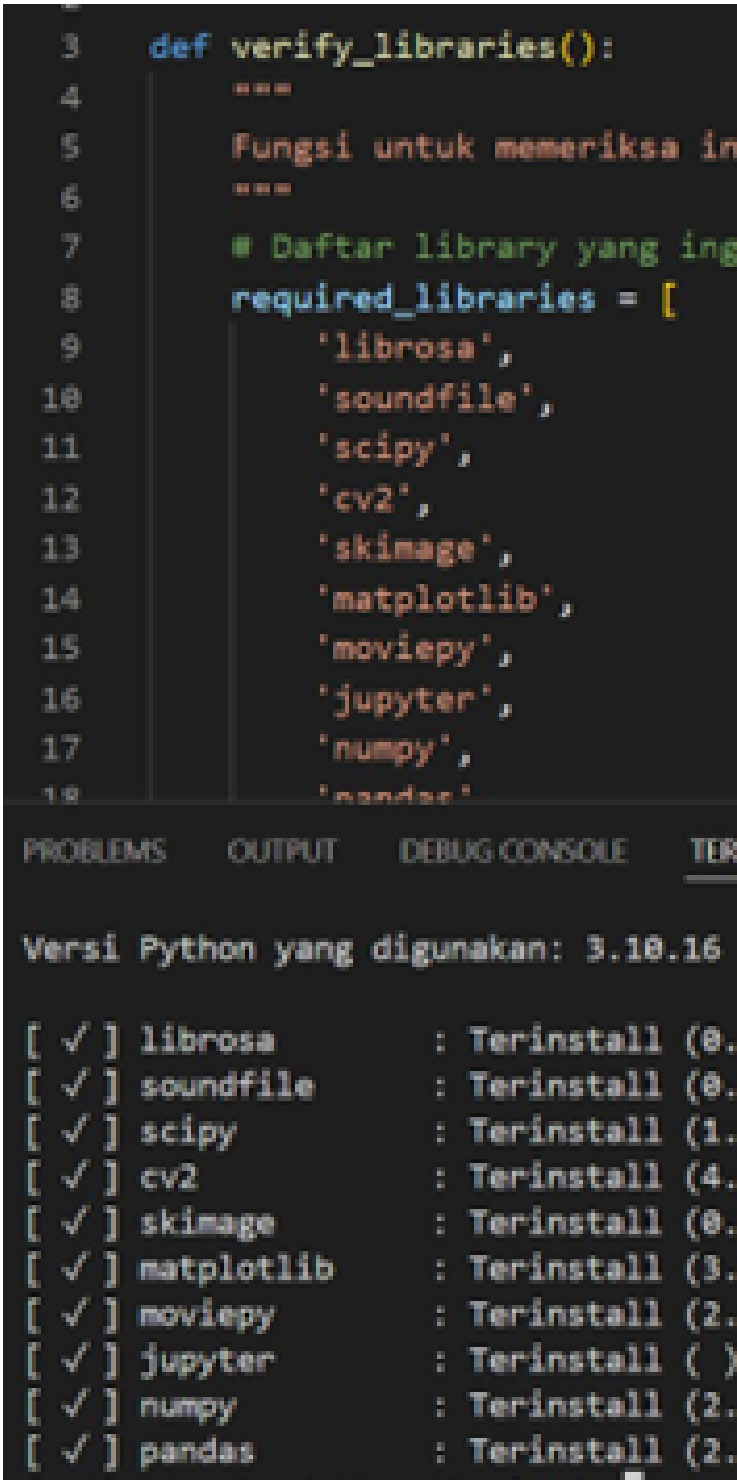
- Daftar library yang berhasil diinstall dengan versinya

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:

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

```
import numpy as np
import matplotlib.pyplot as plt

```

```
# Generate simple sine wave
duration = 2  # seconds
sample_rate = 44100
frequency = 440  # A4 note

t = np.linspace(0, duration, int(sample_rate * duration))
audio_signal = np.sin(2 * np.pi * frequency * t)

# Plot waveform
plt.figure(figsize=(10, 4))
plt.plot(t[:1000], audio_signal[:1000])  # Plot first 1000 samples
plt.title('Sine Wave (440 Hz)')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.grid(True)
plt.savefig('sine_wave_test.png', dpi=150, bbox_inches='tight')
plt.show()

print(f"Generated {duration}s sine wave at {frequency}Hz")
print(f"Sample rate: {sample_rate}Hz")
print(f"Total samples: {len(audio_signal)}")
```

Kode 8: Test audio processing sederhana

3.5.2 Test Image Processing

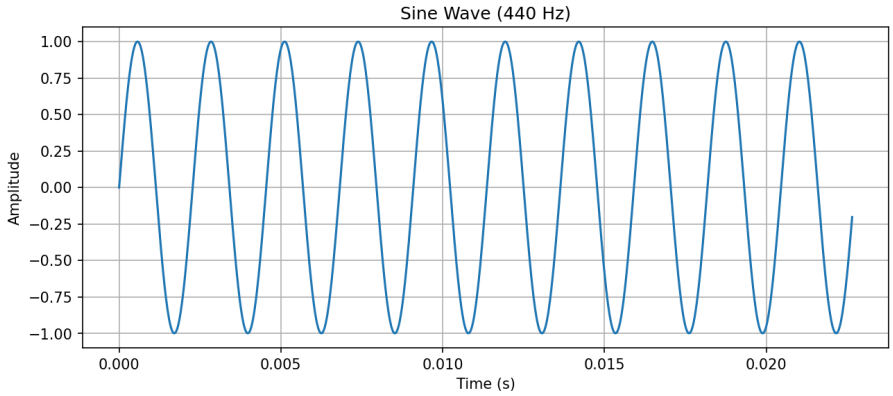
```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image

# Create a simple test image
width, height = 400, 300
image = np.zeros((height, width, 3), dtype=np.uint8)

# Add some patterns
image[:, :width//3, 0] = 255  # Red section
image[:, width//3:2*width//3, 1] = 255  # Green section
image[:, 2*width//3:, 2] = 255  # Blue section

# Add a white circle in the center
center_x, center_y = width//2, height//2
radius = 50
Y, X = np.ogrid[:height, :width]
mask = (X - center_x)**2 + (Y - center_y)**2 <= radius**2
image[mask] = [255, 255, 255]

# Display and save
plt.figure(figsize=(8, 6))
plt.imshow(image)
plt.title('Test Image with RGB Stripes and White Circle')
plt.axis('off')
plt.savefig('test_image.png', dpi=150, bbox_inches='tight')
plt.show()

print(f"Created test image: {width}x{height} pixels")
print(f"Image shape: {image.shape}")
print(f"Image dtype: {image.dtype}")
```

Kode 9: Test image processing sederhana

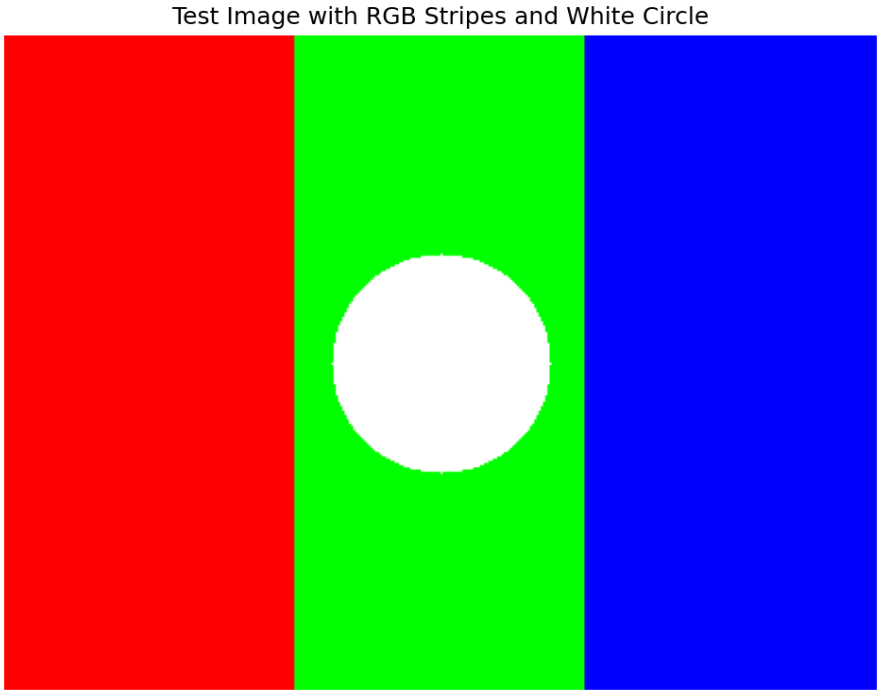
Dokumentasikan hasil eksekusi:

- Screenshot output dari kedua script di atas

```
(multimedia) PS C:\Users\Lenovo\Downloads\Worksheet 1\Worksheet 1> & C:/Users/Lenovo/AppData/Local/Programs/Python/Pytho
10/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:

```
Audio Signal :
Generated 2s sine wave at 440Hz
Sample rate: 44100Hz
Total samples: 88200

Image :
Created test image: 400x300 pixels
Image shape: (300, 400, 3)
```

```
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

```
conda env export > environment.yml
```

Kode 11: Export conda environment

5.2 Untuk venv/uv

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

Kode 12: Export pip requirements

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

```
[name: multimedia
channels:
  - conda-forge
  - defaults
dependencies:
  - _libavif_api=1.3.0=h57928b3_2
  - anyio=4.10.0=pyhe01879c_0
  - aom=3.9.1=he0c23c2_0
  - argon2-cffi=25.1.0=pyhd8ed1ab_0
  - argon2-cffi-bindings=25.1.0=py311h3485c13_0
  - arrow=1.3.0=pyhd8ed1ab_1
  - asttokens=3.0.0=pyhd8ed1ab_1
  - async-lru=2.0.5=pyh29332c3_0
  - attrs=25.3.0=pyh71513ae_0
  - audioread=3.0.1=py311h1ea47a8_2
  - babel=2.17.0=pyhd8ed1ab_0
  - beautifulsoup4=4.13.5=pyha770c72_0
  - bleach=6.2.0=pyh29332c3_4
  - bleach-with-css=6.2.0=h82add2a_4
  - blosc=1.21.6=h4190f5b_0
  - brotli=1.0.9=hcfcfb64_9
  - brotli-bin=1.0.9=hcfcfb64_9
  - brotli-python=1.0.9=py311h12c1d0e_9
  - bzip2=1.0.8=h2bbff1b_6
  - ca-certificates=2025.8.3=h4c7d964_0
  - cached-property=1.5.2=hd8ed1ab_1
  - cached_property=1.5.2=pyha770c72_1
  - cairo=1.18.4=he9e932c_0
  - certifi=2025.8.3=pyhd8ed1ab_0
  - cffi=1.17.1=py311he736701_0
  - cfitsio=3.470=h0af3d06_7
  - charls=2.2.0=h39d44d4_0
  - charset-normalizer=3.4.3=pyhd8ed1ab_0
  - colorama=0.4.6=pyhd8ed1ab_1
  - comm=0.2.3=pyhe01879c_0
  - contourpy=1.3.3=py311h3fd045d_1
```

```
  - cpython=3.11.13=py311hd8ed1ab_0
  - cycler=0.12.1=pyhd8ed1ab_1
  - dav1d=1.2.1=hcfcfb64_0
  - debugpy=1.8.16=py311h5dfdfe8_0
  - decorator=5.2.1=pyhd8ed1ab_0
  - defusedxml=0.7.1=pyhd8ed1ab_0
  - eigen=3.4.0=h91493d7_0
  - exceptiongroup=1.3.0=pyhd8ed1ab_0
  - executing=2.2.0=pyhd8ed1ab_0
  - expat=2.7.1=h8ddb27b_0
  - fontconfig=2.14.1=hb33846d_3
  - fonttools=4.59.2=py311h3f79411_0
  - fqdn=1.5.1=pyhd8ed1ab_1
  - freeglut=3.4.0=h8a1e904_1
  - freetype=2.13.3=h0620614_0
  - fribidi=1.0.10=h8d14728_0
  - gflags=2.2.2=he0c23c2_1005
  - giflib=5.2.2=h64bf75a_0
  - glog=0.5.0=h4797de2_0
  - graphite2=1.3.14=hac47afa_2
  - gst-plugins-base=1.24.12=h91a6125_1
  - gstreamer=1.24.12=hfb93a4f_1
  - gstreamer-orc=0.4.41=h1f81b68_0
  - h11=0.16.0=pyhd8ed1ab_0
  - h2=4.3.0=pyhcf101f3_0
  - harfbuzz=10.2.0=he2f9f60_1
  - hdf5=1.14.5=ha36df97_2
  - hpack=4.1.0=pyhd8ed1ab_0
  - httpcore=1.0.9=pyh29332c3_0
  - httpx=0.28.1=pyhd8ed1ab_0
  - hyperframe=6.1.0=pyhd8ed1ab_0
  - icc_rt=2022.1.0=h6049295_2
  - icu=73.2=h63175ca_0
  - idna=3.10=pyhd8ed1ab_1
  - imagecodecs=2024.9.22=py311hf5f7160_2
  - imageio=2.37.0=pyhfb79c49_0
  - importlib-metadata=8.7.0=pyhe01879c_1
  - ipykernel=6.30.1=pyh3521513_0
  - ipython=9.5.0=pyh6be1c34_0
  - ipython_pygments_lexers=1.1.1=pyhd8ed1ab_0
  - ipywidgets=8.1.7=pyhd8ed1ab_0
  - isoduration=20.11.0=pyhd8ed1ab_1
  - jedi=0.19.2=pyhd8ed1ab_1
  - jinja2=3.1.6=pyhd8ed1ab_0
  - joblib=1.5.2=pyhd8ed1ab_0
  - jpeg=9e=hcfcfb64_3
  - json5=0.12.1=pyhd8ed1ab_0
  - jsonpointer=3.0.0=py311h1ea47a8_1
  - jsonschema=4.25.1=pyhe01879c_0
  - jsonschema-specifications=2025.4.1=pyh29332c3_0
  - jsonschema-with-format-nongpl=4.25.1=he01879c_0
  - jupyter=1.1.1=pyhd8ed1ab_1
  - jupyter-lsp=2.3.0=pyhcf101f3_0
  - jupyter_client=8.6.3=pyhd8ed1ab_1
  - jupyter_console=6.6.3=pyhd8ed1ab_1
  - jupyter_core=5.8.1=pyh5737063_0
  - jupyter_events=0.12.0=pyh29332c3_0
  - jupyter_server=2.17.0=pyhcf101f3_0
  - jupyter_server_terminals=0.5.3=pyhd8ed1ab_1
  - jupyterlab=4.4.6=pyhd8ed1ab_0
  - jupyterlab_pygments=0.3.0=pyhd8ed1ab_2
  - jupyterlab_server=2.27.3=pyhd8ed1ab_1
```

```
  - jupyterlab_widgets=3.0.15=pyhd8ed1ab_0
  - kiwisolver=1.4.9=py311h275cad7_1
  - krb5=1.21.3=hdf4eb48_0
  - lame=3.100=hcfcfb64_1003
  - lark=1.2.2=pyhd8ed1ab_1
  - lazy-loader=0.4=pyhd8ed1ab_2
  - lazy_loader=0.4=pyhd8ed1ab_2
  - lcms2=2.16=h62be587_1
  - lerc=4.0.0=h6470a55_1
  - libabseil=20250127.0=cxx17_h4eb7d71_0
  - libaec=1.1.4=h20038f6_0
  - libavif=1.3.0=he916da2_2
  - libavif16=1.3.0=he916da2_2
  - libblas=3.9.0=34_h5709861_mkl
  - libbrotlicommon=1.0.9=hcfcfb64_9
  - libbrotlidec=1.0.9=hcfcfb64_9
  - libbrotlienc=1.0.9=hcfcfb64_9
  - libcblas=3.9.0=34_h2a3cdd5_mkl
  - libclang13=14.0.6=default_h8e68704_2
  - libdeflate=1.22=h2466b09_0
  - libffi=3.4.4=hd77b12b_1
  - libflac=1.4.3=h63175ca_0
  - libglib=2.84.2=h405b238_0
  - libhwloc=2.12.1=default_h88281d1_1000
  - libiconv=1.18=hc1393d2_2
  - libkrb5=1.21.3=h885b0b7_4
  - liblapack=3.9.0=34_hf9ab0e9_mkl
  - libogg=1.3.5=h2466b09_1
  - libopus=1.5.2=h2466b09_0
  - libpng=1.6.39=h8cc25b3_0
  - libpq=17.4=h4a159e6_2
  - libprotobuf=5.29.3=h65a231f_1
  - librosa=0.11.0=pyhd8ed1ab_0
  - libsndfile=1.2.2=h81429f1_1
  - libsodium=1.0.20=hc70643c_0
  - libtheora=1.2.0=habf309d_0
  - libtiff=4.7.0=h404307b_0
  - libvorbis=1.3.7=h5112557_2
  - libwebp-base=1.6.0=h4d5522a_0
  - libwinpthread=12.0.0.r4.gg4f2fc60ca=h57928b3_9
  - libxml2=2.13.8=h866ff63_0
  - libxslt=1.1.43=h25c3957_0
  - libzopfli=1.0.3=h0e60522_0
  - llvm-openmp=20.1.8=hfa2b4ca_2
  - llvmlite=0.44.0=py311h8b1c7eb_1
  - lz4-c=1.9.4=hcfcfb64_0
  - markupsafe=3.0.2=py311h5082efb_1
  - matplotlib=3.10.5=py311h1ea47a8_0
  - matplotlib-base=3.10.5=py311h43afe63_0
  - matplotlib-inline=0.1.7=pyhd8ed1ab_1
  - minizip=4.0.3=hb68bac4_0
  - mistune=3.1.4=pyhcf101f3_0
  - mkl=2024.2.2=h57928b3_16
  - mpg123=1.32.9=h01009b0_0
  - msgpack-python=1.1.1=py311h3257749_0
  - munkres=1.1.4=pyhd8ed1ab_1
  - nbclient=0.10.2=pyhd8ed1ab_0
  - nbconvert-core=7.16.6=pyh29332c3_0
  - nbformat=5.10.4=pyhd8ed1ab_1
  - nest-asyncio=1.6.0=pyhd8ed1ab_1
  - networkx=3.5=pyhe01879c_0
  - notebook=7.4.5=pyhd8ed1ab_0
```

```
  - notebook-shim=0.2.4=pyhd8ed1ab_1
  - numba=0.61.2=py311h7afb941_1
  - numpy=2.2.6=py311h5e411d1_0
  - opencv=4.10.0=py311h28596fa_7
  - openjpeg=2.5.2=h9b5d1b5_1
  - openssl=3.5.2=h725018a_0
  - overrides=7.7.0=pyhd8ed1ab_1
  - packaging=25.0=pyh29332c3_1
  - pandas=2.3.2=py311h11fd7f3_0
  - pandocfilters=1.5.0=pyhd8ed1ab_0
  - parso=0.8.5=pyhcf101f3_0
  - pcre2=10.42=h0ff8eda_1
  - pickleshare=0.7.5=pyhd8ed1ab_1004
  - pillow=11.3.0=py311hb328d1f_0
  - pip=25.1=pyhc872135_2
  - pixman=0.46.4=h5112557_1
  - platformdirs=4.4.0=pyhcf101f3_0
  - pooch=1.8.2=pyhd8ed1ab_3
  - prometheus_client=0.22.1=pyhd8ed1ab_0
  - prompt-toolkit=3.0.52=pyha770c72_0
  - prompt_toolkit=3.0.52=hd8ed1ab_0
  - psutil=7.0.0=py311h3485c13_1
  - pure_eval=0.2.3=pyhd8ed1ab_1
  - pycparser=2.22=pyh29332c3_1
  - pygments=2.19.2=pyhd8ed1ab_0
  - pyparsing=3.2.3=pyhe01879c_2
  - pyside6=6.7.3=py311h28b127d_1
  - pysocks=1.7.1=pyh09c184e_7
  - python=3.11.13=h981015d_0
  - python-dateutil=2.9.0.post0=pyhe01879c_2
  - python-fastjsonschema=2.21.2=pyhe01879c_0
  - python-json-logger=2.0.7=pyhd8ed1ab_0
  - python-tzdata=2025.2=pyhd8ed1ab_0
  - python_abi=3.11=2_cp311
  - pytz=2025.2=pyhd8ed1ab_0
  - pywavelets=1.9.0=py311h17033d2_1
  - pywin32=311=py311hefeebc8_1
  - pywinpty=2.0.15=py311hda3d55a_0
  - pyyaml=6.0.2=py311h5082efb_2
  - pyzmq=27.0.2=py311hb77b9c8_2
  - qtbase=6.7.3=hd088775_4
  - qtdeclarative=6.7.3=h885b0b7_1
  - qtshadertools=6.7.3=h885b0b7_1
  - qtsvg=6.7.3=h9d4b640_1
  - qttools=6.7.3=hcb596f7_1
  - qtwebchannel=6.7.3=h885b0b7_1
  - qtwebengine=6.7.3=h3869032_1
  - qtwebsockets=6.7.3=h885b0b7_1
  - rav1e=0.7.1=ha073cba_3
  - referencing=0.36.2=pyh29332c3_0
  - requests=2.32.5=pyhd8ed1ab_0
  - rfc3339-validator=0.1.4=pyhd8ed1ab_1
  - rfc3986-validator=0.1.1=pyh9f0ad1d_0
  - rfc3987-syntax=1.1.0=pyhe01879c_1
  - rpds-py=0.27.1=py311hf51aa87_0
  - scikit-image=0.25.2=py311hcf9f919_0
  - scikit-learn=1.7.1=py311h8a15ebc_0
  - scipy=1.16.1=py311h9a1c30b_1
  - send2trash=1.8.3=pyh5737063_1
  - setuptools=78.1.1=py311haa95532_0
  - six=1.17.0=pyhe01879c_1
  - snappy=1.2.2=h7fa0ca8_0
```

```
  - sniffio=1.3.1=pyhd8ed1ab_1
  - soupsieve=2.8=pyhd8ed1ab_0
  - soxr=0.1.3=hcfcfb64_3
  - soxr-python=0.5.0.post1=py311hda3d55a_1
  - sqlite=3.50.2=hda9a48d_1
  - stack_data=0.6.3=pyhd8ed1ab_1
  - standard-aifc=3.13.0=py311h1ea47a8_2
  - standard-sunau=3.13.0=py311h1ea47a8_2
  - svt-av1=3.1.2=hac47afa_0
  - tbb=2021.13.0=h18a62a1_3
  - terminado=0.18.1=pyh5737063_0
  - threadpoolctl=3.6.0=pyhecae5ae_0
  - tifffile=2024.12.12=pyhd8ed1ab_0
  - tinycss2=1.4.0=pyhd8ed1ab_0
  - tk=8.6.15=hf199647_0
  - tomli=2.2.1=pyhe01879c_2
  - tornado=6.5.2=py311h3485c13_0
  - traitlets=5.14.3=pyhd8ed1ab_1
  - types-python-dateutil=2.9.0.20250822=pyhd8ed1ab_0
  - typing-extensions=4.15.0=h396c80c_0
  - typing_extensions=4.15.0=pyhcf101f3_0
  - typing_utils=0.1.0=pyhd8ed1ab_1
  - tzdata=2025b=h04d1e81_0
  - ucrt=10.0.22621.0=haa95532_0
  - unicodedata2=16.0.0=py311h3485c13_1
  - uri-template=1.3.0=pyhd8ed1ab_1
  - urllib3=2.5.0=pyhd8ed1ab_0
  - vc=14.3=h2df5915_10
  - vc14_runtime=14.44.35208=h4927774_10
  - vs2015_runtime=14.44.35208=ha6b5a95_10
  - wcwidth=0.2.13=pyhd8ed1ab_1
  - webcolors=24.11.1=pyhd8ed1ab_0
  - webencodings=0.5.1=pyhd8ed1ab_3
  - websocket-client=1.8.0=pyhd8ed1ab_1
  - wheel=0.45.1=py311haa95532_0
  - widgetsnbextension=4.0.14=pyhd8ed1ab_0
  - win_inet_pton=1.1.0=pyh7428d3b_8
  - winpty=0.4.3=4
  - xz=5.6.4=h4754444_1
  - yaml=0.2.5=h6a83c73_3
  - zeromq=4.3.5=ha9f60a1_7
  - zfp=1.0.1=h2f0f97f_3
  - zipp=3.23.0=pyhd8ed1ab_0
  - zlib=1.2.13=h8cc25b3_1
  - zlib-ng=2.0.7=hcfcfb64_0
  - zstandard=0.23.0=py311h3485c13_3
  - zstd=1.5.6=h8880b57_0
  - pip:
      - ffmpeg==1.4
      - ffmpeg-python==0.2.0
      - future==1.0.0
      - pysoundfile==0.9.0.post1
prefix: C:\Users\Lenovo\miniconda3\envs\multimedia
]
```

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 spesfik 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