Program Studi Teknik Informatika
Institut Teknologi Sumatera

Nama: **A. Edwin Krisandika Putra (122140003)** 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

- 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

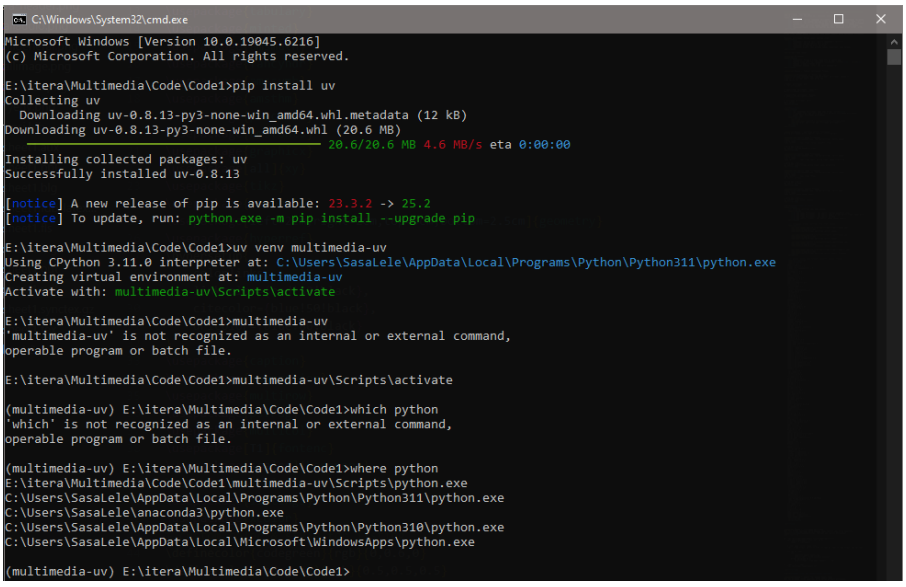
3.2.1 Menggunakan uv (Modern dan cepat)

```
# 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 1: Membuat environment dengan uv

Dokumentasi:

- Tool manajemen environment yang Anda pilih: [**uv**]
- Screenshot atau copy-paste output dari perintah verifikasi environment
- Gambar hasil verifikasi environment:



Gambar 1: Output Installasi dan 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 2: 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 3: 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 4: Instalasi library video

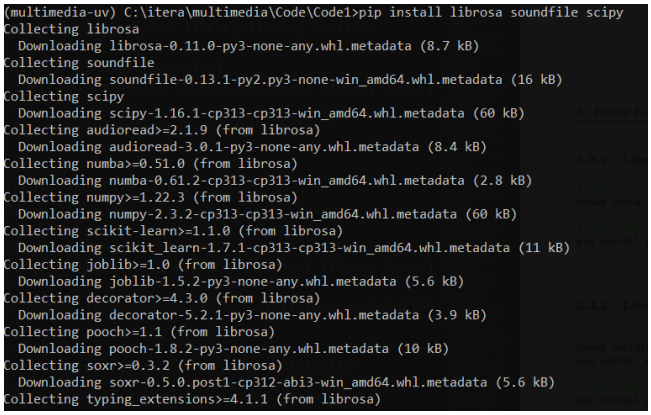
3.3.4 Library General Purpose

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

# Untuk pip (venv/uv):
pip install numpy pandas jupyter
```

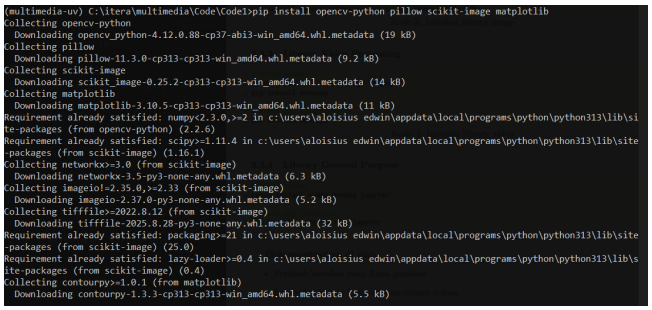
Kode 5: Instalasi library umum

Dokumentasikan di sini:

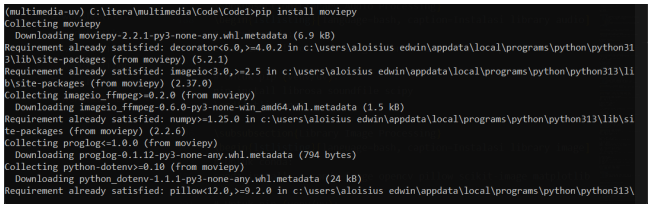
- Perintah instalasi yang Anda gunakan
  - Lib Audio Processing : uv pip install librosa soundfile scipy
  - Lib Image Processing : uv pip install opencv-python pillow scikit-image matplotlib
  - Lib Video Processing : uv pip install moviepy
  - Lib General Purpose : uv pip install numpy pandas jupyter
- Screenshot proses instalasi atau output sukses



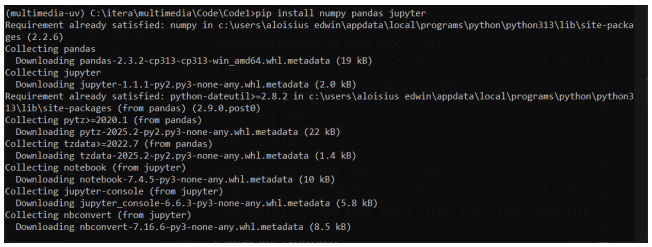
Gambar 2: Proses Instalasi Library Audio



Gambar 3: Proses Instalasi Library Image



Gambar 4: Proses Instalasi Library Video



Gambar 5: Proses Instalasi Library General Purpose

- Daftar library yang berhasil diinstall dengan versinya:

```
anyio==4.10.0
argon2-cffi==25.1.0
argon2-cffi-bindings==25.1.0
arrow==1.3.0
asttokens==3.0.0
async-lru==2.0.5
attrs==25.3.0
audioop-lts==0.2.2
audioread==3.0.1
babel==2.17.0
beautifulsoup4==4.13.5
bleach==6.2.0
certifi==2025.8.3
cffi==1.17.1
charset-normalizer==3.4.3
colorama==0.4.6
comm==0.2.3
contourpy==1.3.3
```

```
cycler==0.12.1
debugpy==1.8.16
decorator==5.2.1
defusedxml==0.7.1
executing==2.2.0
fastjsonschema==2.21.2
fonttools==4.59.2
fqdn==1.5.1
h11==0.16.0
httpcore==1.0.9
httpx==0.28.1
idna==3.10
imageio==2.37.0
imageio-ffmpeg==0.6.0
ipykernel==6.30.1
ipython==9.4.0
ipython_pygments_lexers==1.1.1
ipywidgets==8.1.7
isoduration==20.11.0
jedi==0.19.2
Jinja2==3.1.6
joblib==1.5.2
json5==0.12.1
jsonpointer==3.0.0
jsonschema==4.25.1
jsonschema-specifications==2025.4.1
jupyter==1.1.1
jupyter-console==6.6.3
jupyter-events==0.12.0
jupyter-lsp==2.3.0
jupyter_client==8.6.3
jupyter_core==5.8.1
jupyter_server==2.17.0
jupyter_server_terminals==0.5.3
jupyterlab==4.4.6
jupyterlab_pygments==0.3.0
jupyterlab_server==2.27.3
jupyterlab_widgets==3.0.15
kiwisolver==1.4.9
lark==1.2.2
lazy_loader==0.4
librosa==0.11.0
llvmlite==0.44.0
MarkupSafe==3.0.2
matplotlib==3.10.5
matplotlib-inline==0.1.7
mistune==3.1.3
moviepy==2.2.1
msgpack==1.1.1
nbclient==0.10.2
nbconvert==7.16.6
nbformat==5.10.4
nest-asyncio==1.6.0
networkx==3.5
notebook==7.4.5
notebook_shim==0.2.4
numba==0.61.2
numpy==2.2.6
opencv-python==4.12.0.88
packaging==25.0
pandas==2.3.2
pandocfilters==1.5.1
parso==0.8.5
pillow==11.3.0
platformdirs==4.4.0
pooch==1.8.2
proglog==0.1.12
prometheus_client==0.22.1
prompt_toolkit==3.0.52
psutil==7.0.0
pure_eval==0.2.3
pycparser==2.22
Pygments==2.19.2
pyparsing==3.2.3
python-dateutil==2.9.0.post0
python-dotenv==1.1.1
python-json-logger==3.3.0
pytz==2025.2
pywin32==311
pywinpty==3.0.0
PyYAML==6.0.2
pyzmq==27.0.2
referencing==0.36.2
requests==2.32.5
rfc3339-validator==0.1.4
rfc3986-validator==0.1.1
rfc3987-syntax==1.1.0
rpds-py==0.27.1
scikit-image==0.25.2
scikit-learn==1.7.1
scipy==1.16.1
Send2Trash==1.8.3
setuptools==80.9.0
six==1.17.0
sniffio==1.3.1
soundfile==0.13.1
soupsieve==2.8
soxr==0.5.0.post1
stack-data==0.6.3
standard-aifc==3.13.0
```

```
standard-chunk==3.13.0
standard-sunau==3.13.0
terminado==0.18.1
threadpoolctl==3.6.0
tifffile==2025.8.28
tinycss2==1.4.0
tornado==6.5.2
tqdm==4.67.1
traitlets==5.14.3
types-python-dateutil==2.9.0.20250822
typing_extensions==4.15.0
tzdata==2025.2
uri-template==1.3.0
urllib3==2.5.0
uv==0.8.13
wcwidth==0.2.13
webcolors==24.11.1
webencodings==0.5.1
websocket-client==1.8.0
widgetsnbextension==4.0.14
```

3.4 Bagian 3: Verifikasi Instalasi

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

```python
import importlib

def test_libraries(requirements_file="requirements.txt"):
    with open(requirements_file, "r") as f:
        lines = f.readlines()

    # ambil nama library tanpa versi
    libs = [line.strip().split("==")[0] for line in lines if line.strip() and not line.startswith("
    #")]

    for lib in libs:
        try:
            importlib.import_module(lib)
            print(f"[OK]   {lib} berhasil diimport")
        except Exception as e:
            print(f"[FAIL] {lib} gagal diimport -> {e}")

if __name__ == "__main__":
    test_libraries("requirements.txt")
```

Kode 6: Uji library multimedia

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

```python
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 7: 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 8: Test image processing sederhana

Dokumentasikan hasil eksekusi:

- Screenshot output dari kedua script di atas
- Gambar yang dihasilkan (sine_wave_test.png dan test_image.png)
- Error message jika ada dan cara mengatasinya

4 Bagian Laporan

4.1 Output Verifikasi Instalasi

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

```
[PASTE OUTPUT DI SINI]
```

Kode 9: 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?**
   [Jawaban Anda di sini]
2. **Apa perbedaan utama antara conda, venv, dan uv? Mengapa Anda memilih tool yang Anda gunakan?**
   [Jawaban Anda di sini]
3. **Library mana yang paling sulit diinstall dan mengapa?**
   [Jawaban Anda di sini]
4. **Bagaimana cara mengatasi masalah dependency conflict jika terjadi?**
   [Jawaban Anda di sini]
5. **Jelaskan fungsi dari masing-masing library yang berhasil Anda install!**
   [Jawaban Anda di sini]

4.4 Troubleshooting

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

- **Masalah 1:** *[Deskripsi masalah]*
  Solusi: *[Cara mengatasi]*
- **Masalah 2:** *[Deskripsi masalah]*
  Solusi: *[Cara mengatasi]*

5 Export Environment untuk Reproduksi

Sebagai langkah terakhir, export environment Anda agar dapat direproduksi:

5.1 Untuk Conda

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

Kode 10: Export conda environment

5.2 Untuk venv/uv

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

Kode 11: Export pip requirements

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

```
[PASTE ISI FILE DI SINI]
```

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

[Kesimpulan Anda di sini]

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting.

References