Program Studi Teknik Informatika
Institut Teknologi Sumatera

Nama: **Dzaki Gastiadirrijal (122140030)** 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

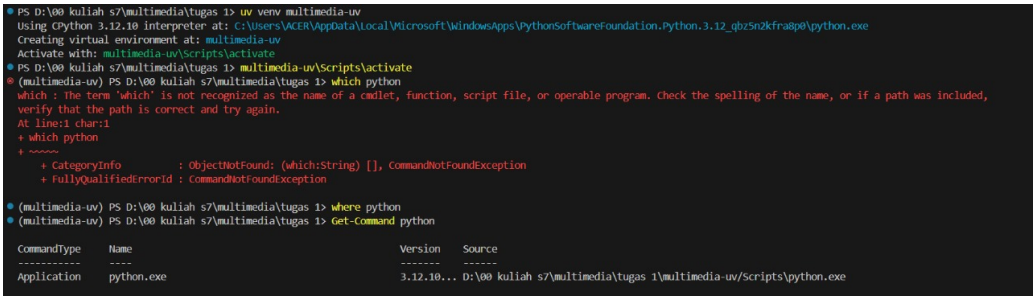
3.2.3 Opsi 3: Menggunakan uv (Modern dan cepat)

```
# 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: [**uv**]
- Screenshot atau copy-paste output dari perintah verifikasi environment



Gambar 1: Hasil verifikasi environment dengan uv (which python tidak bisa digunakan di windows)

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
```

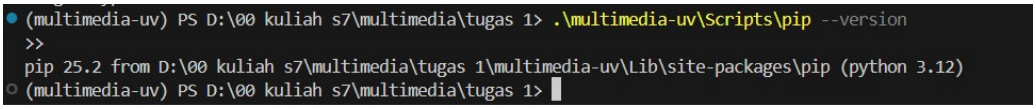
Kode 7: Instalasi library umum

Dokumentasikan di sini:

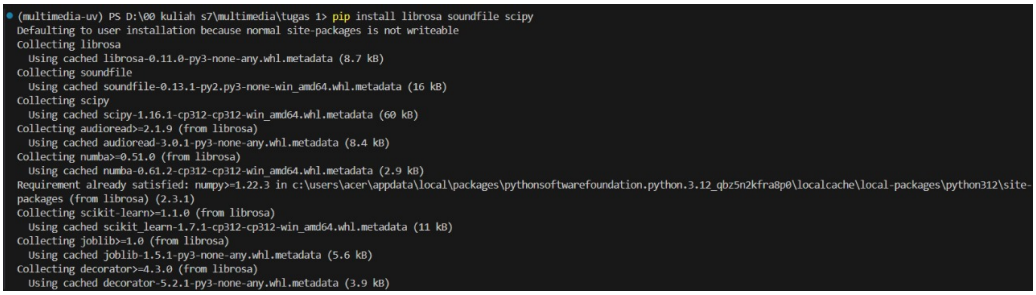
- Perintah instalasi yang Anda gunakan

  1. Untuk mendownload pip dalam environment multimedia-uv `.\multimedia-uv\Scripts\python.exe -m ensurepip -upgrade`
  2. pip install librosa soundfile scipy
  3. pip install opencv-python pillow scikit-image matplotlib
  4. pip install moviepy
  5. pip install numpy pandas jupyter

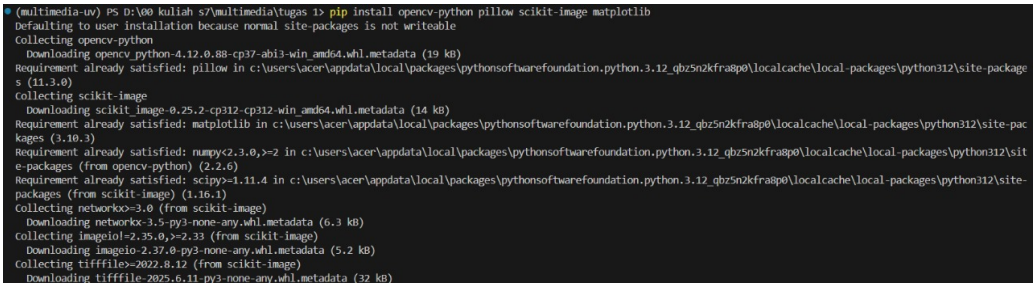
- Screenshot proses instalasi atau output sukses



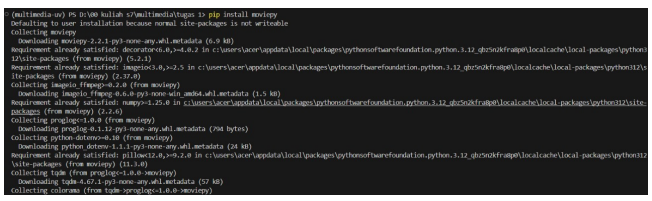
Gambar 2: command untuk verifikasi pip dalam env multimedia-uv



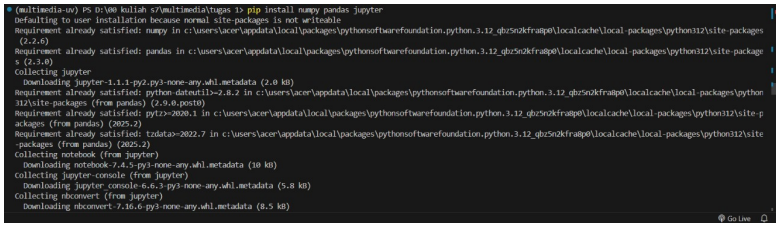
Gambar 3: Proses instalasi librosa soundfile scipy



Gambar 4: Proses instalasi opencv-python pillow scikit-image matplotlib



Gambar 5: Proses instalasi moviepy



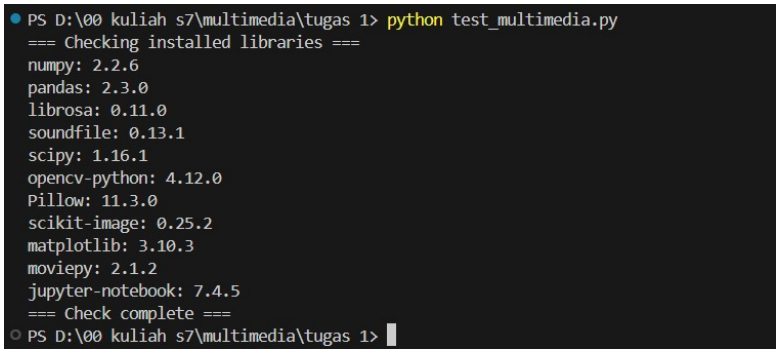
Gambar 6: Proses instalasi numpy pandas jupyter

- Daftar library yang berhasil diinstall dengan versinya

```
numpy: 2.2.6
pandas: 2.3.2
librosa: 0.11.0
soundfile: 0.13.1
scipy: 1.16.1
opencv-python: 4.12.0
Pillow: 11.3.0
scikit-image: 0.25.2
matplotlib: 3.10.5
moviepy: 2.1.2
jupyter-notebook: 7.4.5
```

3.4 Bagian 3: Verifikasi Instalasi

Buat file Python sederhana untuk menguji semua library yang telah diinstall:
Jalankan script dan dokumentasikan hasilnya:



Gambar 7: Output script untuk verifikasi instalasi (script dilampirkan pada bagian 4)

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 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

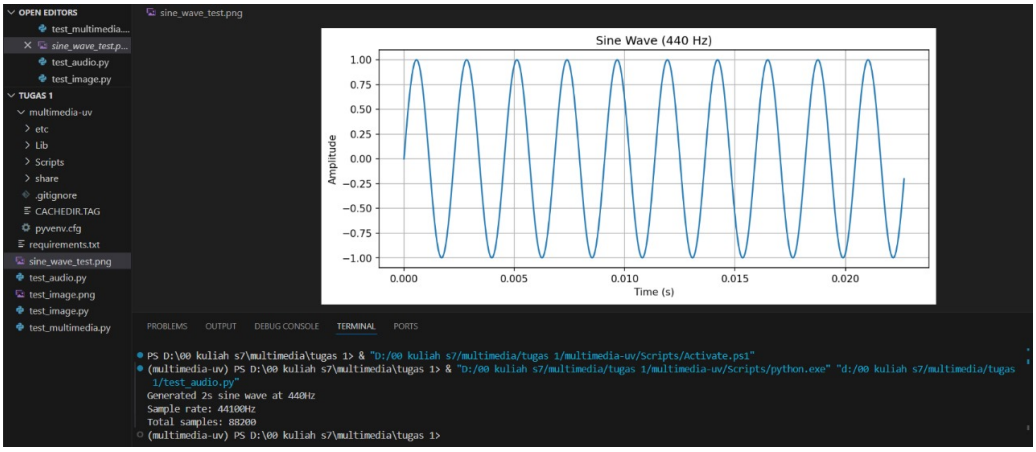
```
(multimedia-uv) PS D:\00 kuliah s7\multimedia\tugas 1> & "D:/00 kuliah s7/multimedia/tugas 1/multimedia-uv/Scripts/python.exe" "d:/00 kuliah s7/multimedia/tugas
 1/test_audio.py"
Generated 2s sine wave at 440Hz
Sample rate: 44100Hz
Total samples: 88200
(multimedia-uv) PS D:\00 kuliah s7\multimedia\tugas 1>
```

Gambar 8: Output script Audio

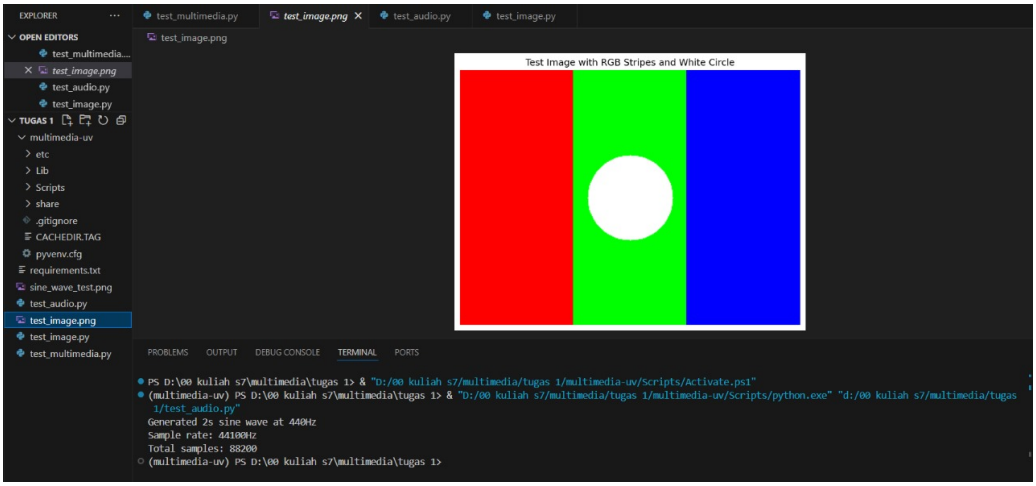
```
(multimedia-uv) PS D:\00 kuliah s7\multimedia\tugas 1> & "D:/00 kuliah s7/multimedia/tugas 1/multimedia-uv/Scripts/python.exe" "d:/00 kuliah s7/multimedia/tugas
 1/test_image.py"
Created test image: 400x300 pixels
Image shape: (300, 400, 3)
Image dtype: uint8
(multimedia-uv) PS D:\00 kuliah s7\multimedia\tugas 1>
```

Gambar 9: Output script Image

- Gambar yang dihasilkan (sine_wave_test.png dan test_image.png)



Gambar 10: Output Gambar Audio



Gambar 11: Output Gambar Image

- Error message jika ada dan cara mengatasinya

4 Bagian Laporan

4.1 Output Verifikasi Instalasi

```python
# check_libs.py
print("=== Checking installed libraries ===")

try:
    import numpy
    print("numpy:", numpy.__version__)
except ImportError:
    print("numpy: NOT INSTALLED")

try:
    import pandas
    print("pandas:", pandas.__version__)
except ImportError:
    print("pandas: NOT INSTALLED")

try:
    import librosa
    print("librosa:", librosa.__version__)
except ImportError:
    print("librosa: NOT INSTALLED")

try:
    import soundfile
    print("soundfile:", soundfile.__version__)
except ImportError:
    print("soundfile: NOT INSTALLED")

try:
    import scipy
    print("scipy:", scipy.__version__)
except ImportError:
    print("scipy: NOT INSTALLED")

try:
    import cv2
    print("opencv-python:", cv2.__version__)
except ImportError:
    print("opencv-python: NOT INSTALLED")

try:
    import PIL
    print("Pillow:", PIL.__version__)
except ImportError:
    print("Pillow: NOT INSTALLED")

try:
    import skimage
    print("scikit-image:", skimage.__version__)
except ImportError:
    print("scikit-image: NOT INSTALLED")

try:
    import matplotlib
    print("matplotlib:", matplotlib.__version__)
except ImportError:
    print("matplotlib: NOT INSTALLED")

try:
    import moviepy
    print("moviepy:", moviepy.__version__)
except ImportError:
    print("moviepy: NOT INSTALLED")
```

```

try:
    import notebook
    print("jupyter-notebook:", notebook.__version__)
except ImportError:
    print("jupyter-notebook: NOT INSTALLED")

print("=== Check complete ===")
```

Kode 10: Kode Yang dibuat untuk verifikasi instalasi di test multimedia py

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

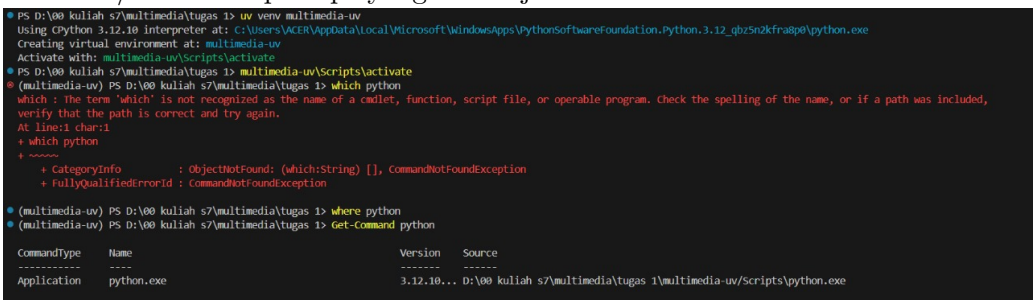
```
[
numpy: 2.2.6
pandas: 2.3.0
librosa: 0.11.0
soundfile: 0.13.1
scipy: 1.16.1
opencv-python: 4.12.0
Pillow: 11.3.0
scikit-image: 0.25.2
matplotlib: 3.10.3
moviepy: 2.1.2
jupyter-notebook: 7.4.5
]
```

Kode 11: Output verifikasi instalasi

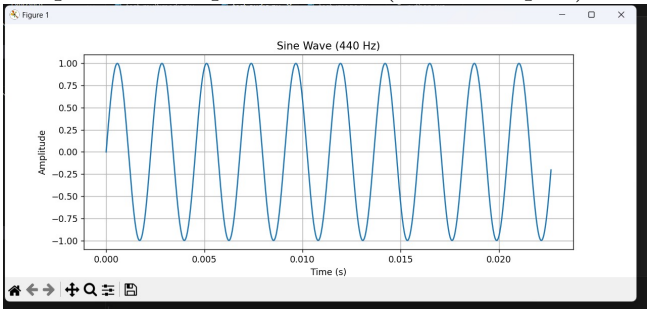
4.2 Screenshot Hasil Test

Sisipkan screenshot atau gambar hasil dari:

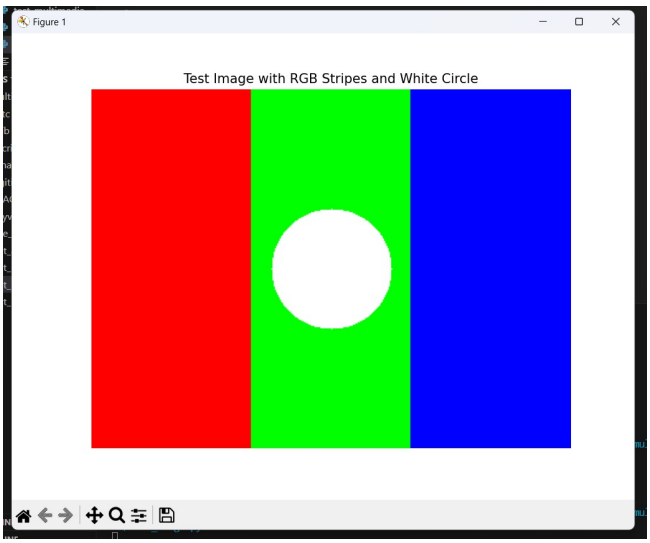
- Terminal/command prompt yang menunjukkan environment aktif



- 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?**

   Menurut saya misalnya project saat ini yang saya ingin library x untuk versi 1.x tetapi project lainnya membutuhkan versi 2.x maka otomatis akan tabrakan, dibuatnya environment sendiri untuk mencegah hal seperti itu untuk terjadi agar mereka masing masing punya library sama dengan versi berbeda.

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

   Alasan jujur utama adalah karena tahapan paling mudah diikuti untuk instalasi bagi saya adalah uv lewat bantuan ChatGPT juga. Dan mudahnya uv adalah hanya membuat env kosong -> lalu tinggal instal dengan pip.

3. **Library mana yang paling sulit diinstall dan mengapa?**

   Sebenarnya untuk proses instalasi menggunakan pip tidak ada yang rumit hanya berbeda waktu instalasinya. mungkin ini dikarenakan menggunakan uv, masalah yang ditemukan adalah instalasi pada environment yang salah.

4. **Bagaimana cara mengatasi masalah dependency conflict jika terjadi?**

   untuk uv, bisa digunakan uv pip check lalu sync requirements.txt untuk disesuaikan. bisa juga menggunakan download versi secara manual atau versi yang diinginkan.

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

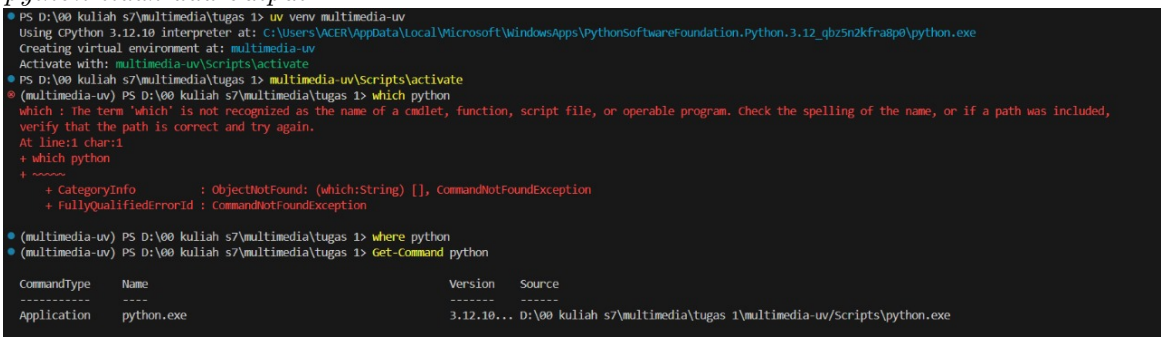
   - **librosa**: analisis audio/musik, baca file suara, ekstrak fitur, dan membuat spectrogram.

- **soundfile**: baca/tulis file audio.
- **scipy**: pemrosesan sinyal dan perhitungan ilmiah tambahan.
- **opencv-python**: computer vision, olah citra/video, deteksi objek, deteksi wajah.
- **Pillow**: menambahkan image processing untuk python.
- **scikit-image**: pemrosesan citra ilmiah (segmentasi, transformasi, edge detection).
- **matplotlib**: visualisasi data/gambar dalam bentuk grafik atau plot.
- **moviepy**: edit video sederhana.
- **numpy**: operasi matematika & array multidimensi.
- **pandas**: olah data tabular.
- **jupyter-notebook**: lingkungan interaktif untuk coding dan visualisasi langsung di browser.

4.4 Troubleshooting

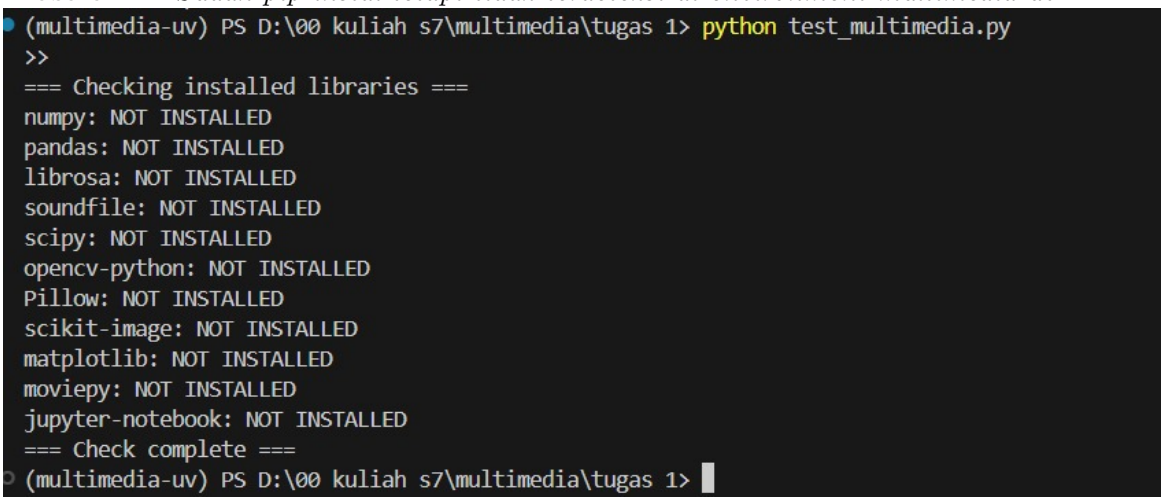
Dokumentasikan masalah yang Anda hadapi (jika ada) dan cara mengatasinya:

- **Masalah 1:** *which python tidak bisa digunakan karena tidak ada di versi windows, dan where python tidak ada output*



Solusi: *Menggunakan where python, lalu menggunakan Command-get python untuk menemukan source python.exe nya*

- **Masalah 2:** *Sudah pip instal tetapi tidak terdeteksi di environment multimedia-uv*



Solusi: *Gunakan command .\multimedia-uv\Scripts\python.exe -m ensurepip –upgrade untuk menginstall pip ke dalam folder environment yang benar.*

5 Export Environment untuk Reproduksi

Sebagai langkah terakhir, export environment Anda agar dapat direproduksi:

5.1 Untuk Conda

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

Kode 12: Export conda environment

5.2 Untuk venv/uv

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

Kode 13: Export pip requirements

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

```
    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
    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==4.59.1
    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==1.5.1
    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==2.2.6
    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==4.3.8
    pooch==1.8.2
    proglog==0.1.12
    prometheus_client==0.22.1
    prompt_toolkit==3.0.51
    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==0.27.0
```

```
    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==2.7
    soxr==0.5.0.post1
    stack-data==0.6.3
    terminado==0.18.1
    threadpoolctl==3.6.0
    tifffile==2025.6.11
    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
    wcwidth==0.2.13
    webcolors==24.11.1
    webencodings==0.5.1
    websocket-client==1.8.0
    widgetsnbextension==4.0.14
```

Kode 14: Environment/Requirements file

6 Kesimpulan

Tuliskan kesimpulan Anda mengenai:

- Pengalaman setup Python environment untuk multimedia
  Proses setup ternyata lebih kompleks dari yang dibayangkan. Meskipun sebagian besar library dapat diinstal dengan mudah menggunakan uv, tetap ada beberapa hal teknis yang perlu diperhatikan supaya environmentnya berjalan stabil.
- Persiapan untuk project multimedia selanjutnya
  Untuk persiapan project multimedia berikutnya, saya berencana fokus pada augmented reality (AR). Rencana saya adalah mencari referensi fitur-fitur AR yang pernah dibuat sebelumnya.
- Saran untuk mahasiswa lain yang akan melakukan setup serupa
  Jangan ragu menonton tutorial di YouTube atau meminta bantuan dari teman yang sudah bisa melakukannya.

7 Referensi

Berikut referensi yang digunakan selama proses setup dan troubleshooting Python environment:

- ChatGPT - setup uv dan membuat test multimedia py
- ChatGPT - Troubleshooting "which python" yang tidak bisa bekerja
- ChatGPT - Troubleshooting isu instalasi Python/UV environment

References