



Nama: **M.Raihan Athalah Ilham (122140022)**
Environment untuk Multimedia

Tugas Ke: **Worksheet 1: Setup Python**

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: [Membuat Enviroment dengan uv]
- 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

```
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
- Screenshot proses instalasi atau output sukses
- Daftar library yang berhasil diinstall dengan versinya

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

```

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

4 Bagian Laporan

4.1 Output Verifikasi Instalasi

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

```

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)}")
25
26 RGB
27
28 import numpy as np
29 import matplotlib.pyplot as plt
30 from PIL import Image
31
32 # Create a simple test image
33 width, height = 400, 300
34 image = np.zeros((height, width, 3), dtype=np.uint8)
35
36 # Add some patterns
37 image[:, :width//3, 0] = 255 # Red section
38 image[:, width//3:2*width//3, 1] = 255 # Green section
39 image[:, 2*width//3:, 2] = 255 # Blue section
40

```

```
41 # Add a white circle in the center
42 center_x, center_y = width//2, height//2
43 radius = 50
44 Y, X = np.ogrid[:height, :width]
45 mask = (X - center_x)**2 + (Y - center_y)**2 <= radius**2
46 image[mask] = [255, 255, 255]
47
48 # Display and save
49 plt.figure(figsize=(8, 6))
50 plt.imshow(image)
51 plt.title('Test Image with RGB Stripes and White Circle')
52 plt.axis('off')
53 plt.savefig('test_image.png', dpi=150, bbox_inches='tight')
54 ]
```

Kode 10: Output verifikasi instalasi

4.2 Screenshot Hasil Test

Sisipkan screenshot atau gambar hasil dari:

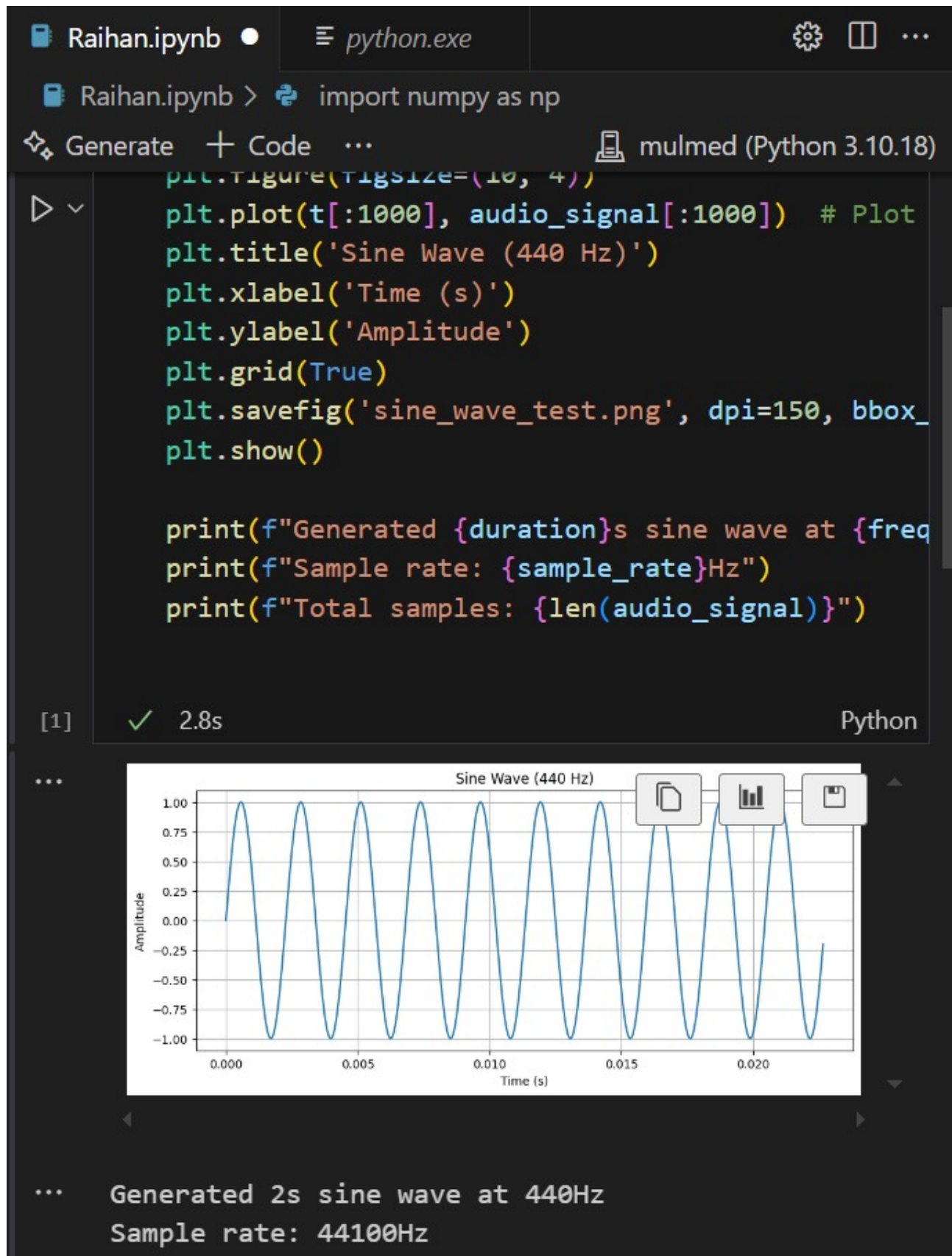
```
C:\Rehan ACER>uv venv multimedia
Using CPython 3.10.18
Creating virtual environment at: multimedia
Activate with: multimedia\Scripts\activate

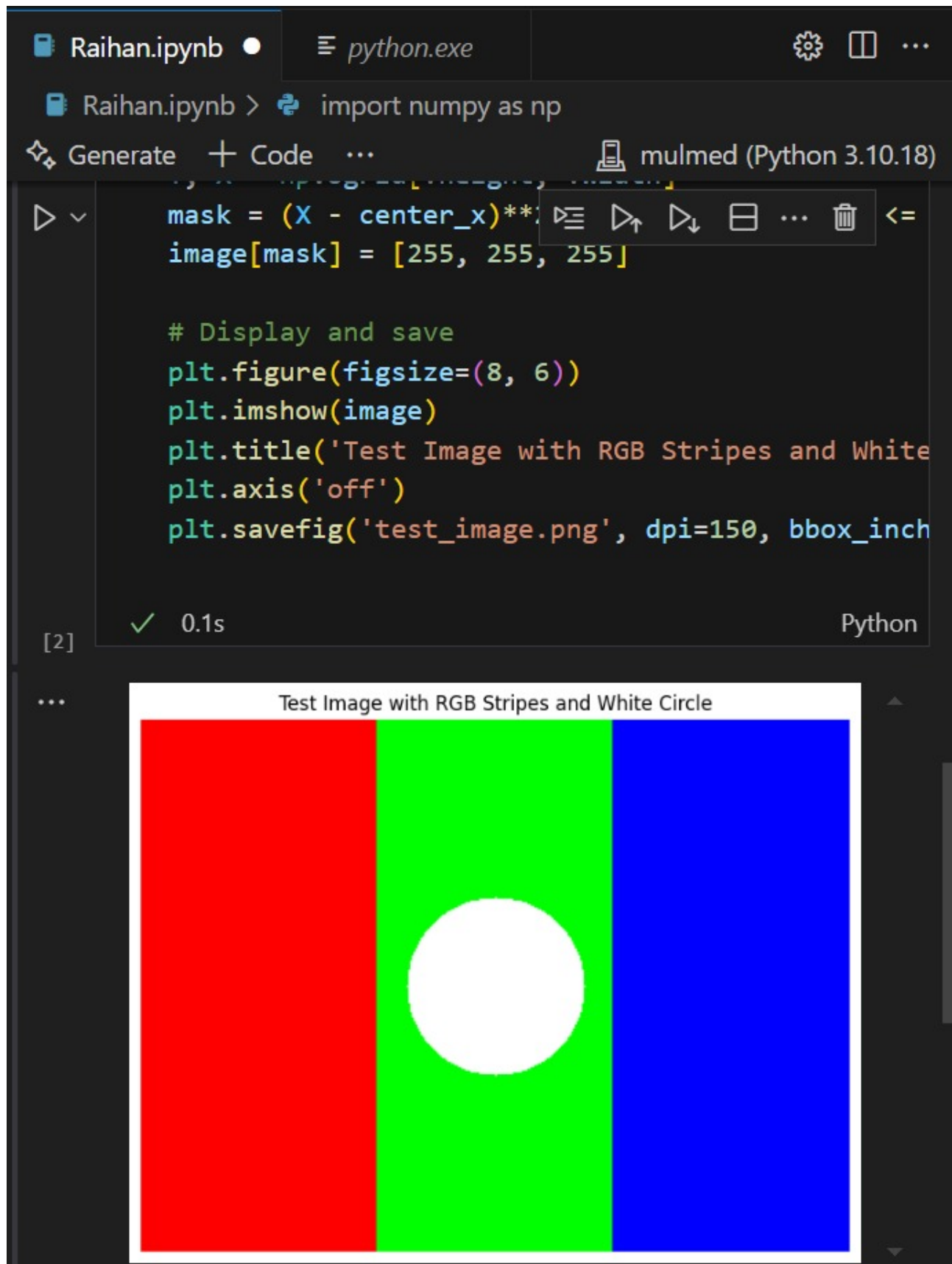
C:\Rehan ACER>multimedia\Scripts\activate

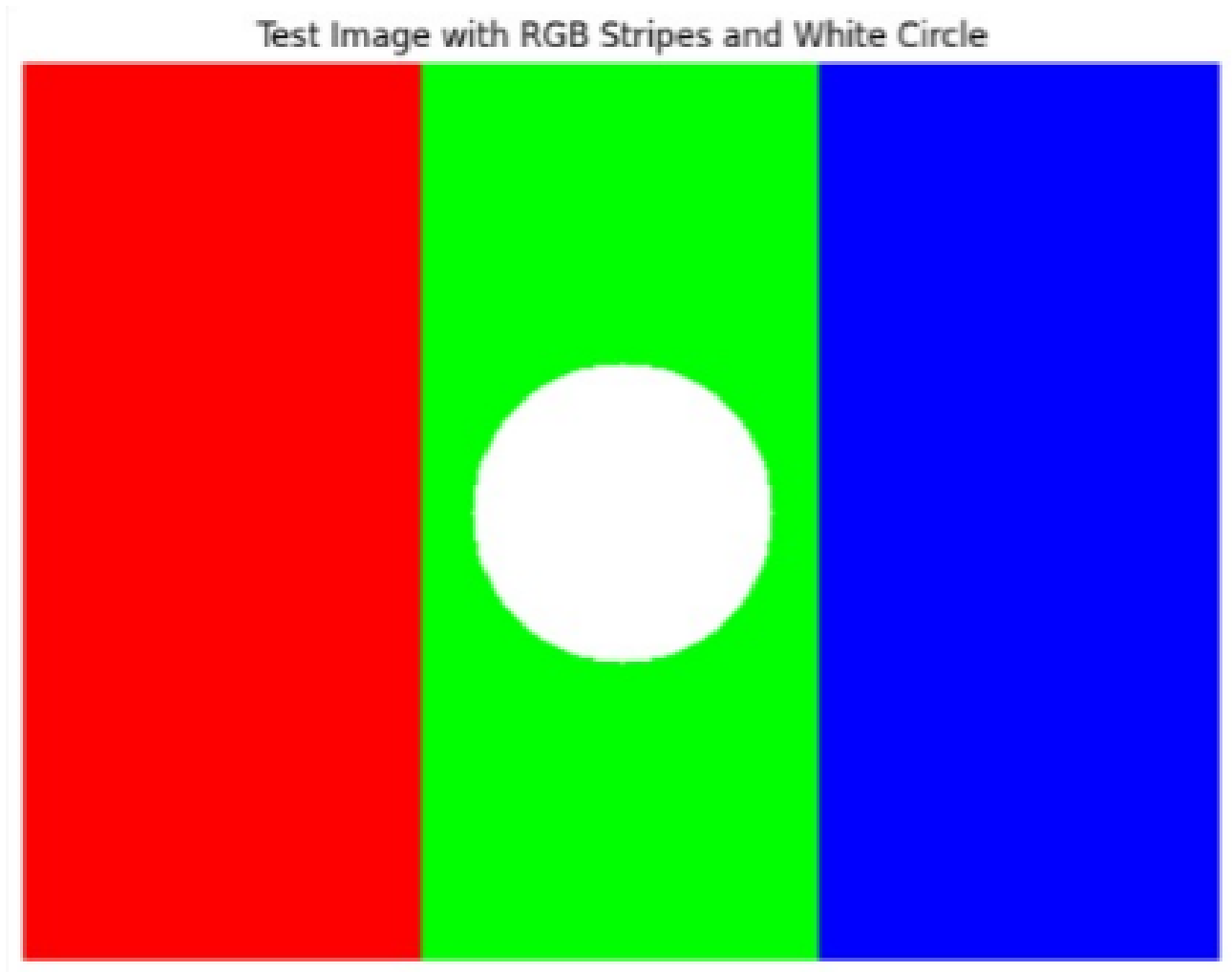
(multimedia) C:\Rehan ACER>uv pip install opencv-python pillow scikit-image matplotlib
Using Python 3.10.18 environment at: multimedia
Resolved 18 packages in 1.11s
Installed 18 packages in 1.47s
+ contourpy==1.3.2
+ cycler==0.12.1
+ fonttools==4.59.2
+ imageio==2.37.0
+ kiwisolver==1.4.9
+ lazy-loader==0.4
+ matplotlib==3.10.5
+ networkx==3.4.2
+ numpy==2.2.6
+ opencv-python==4.12.0.88
+ packaging==25.0
+ pillow==11.3.0
+ pyparsing==3.2.3
+ python-dateutil==2.9.0.post0
+ scikit-image==0.25.2
+ scipy==1.15.3
```

```
(multimedia) C:\Rehan ACER>uv pip install moviepy
Using Python 3.10.18 environment at: multimedia
Resolved 10 packages in 317ms
Installed 7 packages in 123ms
+ colorama==0.4.6
+ decorator==5.2.1
+ imageio-ffmpeg==0.6.0
+ moviepy==2.2.1
+ proglog==0.1.12
+ python-dotenv==1.1.1
+ tqdm==4.67.1
```

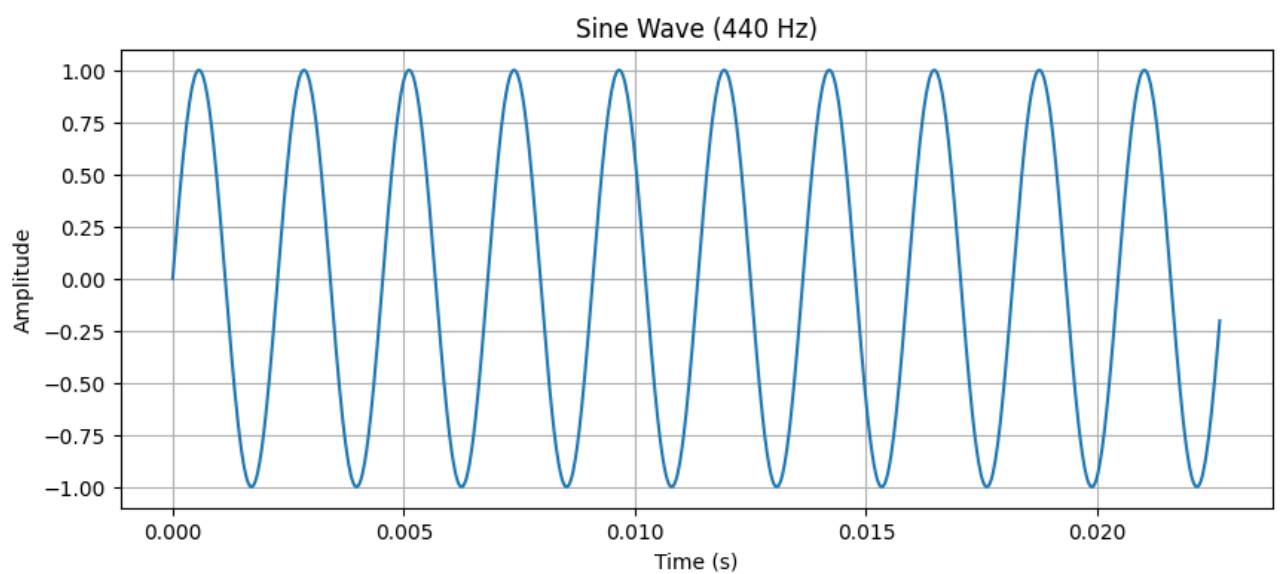
```
(multimedia) C:\Rehan ACER>uv pip install numpy pandas jupyter
Using Python 3.10.18 environment at: multimedia
Resolved 105 packages in 951ms
Installed 99 packages in 3.42s
+ anyio==4.10.0
+ argon2-cffi==25.1.0
+ argon2-cffi-bindings==25.1.0
+ arrow==1.3.0
+ asttokens==3.0.0
+ async-lru==2.0.5
+ attrs==25.3.0
+ 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
+ comm==0.2.3
+ debugpy==1.8.16
+ defusedxml==0.7.1
+ exceptiongroup==1.3.0
+ executing==2.2.0
+ fastjsonschema==2.21.2
+ fqdn==1.5.1
+ h11==0.16.0
+ httpcore==1.0.9
```



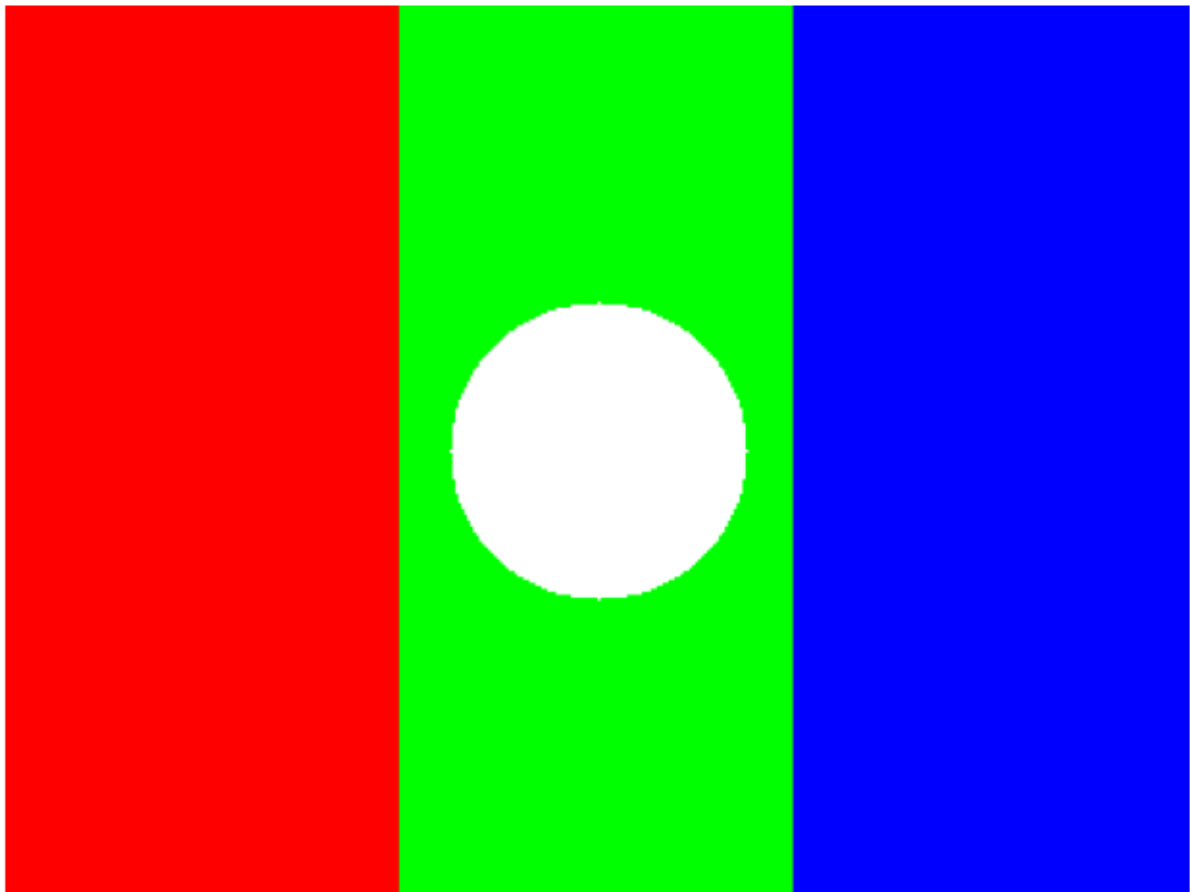


- Terminal/command prompt yang menunjukkan environment aktif
- Output1 dari script test_multimedia.py



- Output2 dari script test_multimedia.py

Test Image with RGB Stripes and White 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, environment terpisah penting supaya project lebih stabil dan nggak tercampur dengan project lain. Apalagi di multimedia, kadang butuh library versi tertentu yang sensitif. Dengan environment sendiri, saya juga bisa bebas eksperimen tanpa takut merusak sistem atau project lain.]
- 2. Apa perbedaan utama antara conda, venv, dan uv? Mengapa Anda memilih tool yang Anda gunakan?**
[Dari yang saya tahu, venv itu bawaan Python dan cukup ringan, tapi kurang fleksibel. conda lebih lengkap karena sudah ada package manager-nya dan cocok buat project yang butuh banyak dependensi, apalagi yang non-Python. Nah, uv lebih modern dan cepat setup-nya, dan saya suka karena praktis. Saya pilih uv karena ingin coba yang ringan dan efisien.]
- 3. Library mana yang paling sulit diinstall dan mengapa?**
[Sejauh ini saya belum nemu yang benar-benar sulit diinstall. Semua berjalan cukup lancar, mungkin karena saya sudah pakai environment yang bersih dan juga installnya pakai tool yang cepat. Tapi saya juga belum sampai tahap install library yang rumit banget atau butuh compile dari source.]

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

[Kalau ada konflik dependency, biasanya saya cek dulu versi yang bentrok. Kadang bisa diakalin dengan upgrade atau downgrade salah satu library. Tapi kalau sudah ribet, biasanya saya bikin environment baru aja biar bersih dan mulai dari awal. Itu cara paling aman menurut saya.]

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

[Librosa digunakan untuk memproses data audio, sedangkan Soundfile berfungsi untuk membaca dan menulis file audio dalam berbagai format. Scipy membantu dalam melakukan komputasi ilmiah, termasuk pengolahan sinyal. Untuk pengolahan citra, OpenCV sangat berguna, sementara Pillow lebih fokus pada manipulasi gambar seperti mengubah ukuran atau format. Scikit-image digunakan untuk analisis gambar secara lebih mendalam. Matplotlib berperan penting dalam membuat visualisasi data. FFmpeg dimanfaatkan untuk memproses video, dan MoviePy digunakan untuk mengedit video secara langsung melalui Python. NumPy menjadi dasar untuk komputasi numerik, sedangkan Pandas memudahkan manipulasi dan analisis data dalam bentuk tabel. Terakhir, Jupyter digunakan sebagai platform interaktif untuk menulis dan menjalankan kode sambil mendokumentasikan prosesnya.]

4.4 Troubleshooting

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

- **Masalah 1:** *[Saat mengikuti template latex dan eror]*

Solusi: *[Menanyakan ke Chatgpt dan gemini bagaimana solusinya, dan solusinya adalah menambahkan usepackage font.]*

- **Masalah 2:** *[Terjadi berbagai komplikasi file dalam setting env pythonnya.]*

Solusi: *[menggunakan github co pilot dan langsung menerapkan solusinya.]*

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 [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)}")
25 ]

```

Kode pi: Environment/Requirements file

```

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 ]

```

[raihan.ipynb](#)

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

[Selama setup environment Python untuk keperluan multimedia, saya belajar pentingnya memilih tool yang tepat dan menjaga agar dependensi tidak saling bertabrakan. Saya menggunakan environment

terpisah agar instalasi library seperti Librosa, OpenCV, atau MoviePy bisa berjalan mulus tanpa mengganggu sistem global. Prosesnya cukup lancar karena saya dibantu teman dan juga memakai tool seperti uv yang cepat dan ringan. Untuk persiapan project multimedia selanjutnya, saya berencana membuat template environment yang sudah berisi library dasar, jadi tidak perlu setup dari nol setiap kali mulai project baru. Ini akan menghemat waktu dan menjaga konsistensi konfigurasi. Bagi mahasiswa lain yang ingin melakukan hal serupa, saya sarankan untuk selalu menggunakan environment terisolasi, mencatat versi library yang digunakan, dan jangan ragu mencoba tool modern yang bisa mempercepat proses setup. Dokumentasi yang rapi juga sangat membantu kalau suatu saat perlu mengulang atau membagikan project ke orang lain.]

7 Referensi

Sertakan referensi yang Anda gunakan selama proses setup dan troubleshooting. [1]

References

- [1] OpenAI, “ChatGPT: Language Model by OpenAI,” <https://openai.com/chatgpt>, 2023, accessed: 2023-10-01.