



Nama: **Rafki Haykhal Alif (122140035)** 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: [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) Microsoft Corporation. All rights reserved.

```

C:\Rafki Mulmed>uv venv multimed
Using CPython 3.10.18
Creating virtual environment at: multimed
Activate with: multimed\Scripts\activate

```

```

C:\Rafki Mulmed>multimed\scripts\activate

```

```

(multimed) C:\Rafki Mulmed>

```

```

+ 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
+ six==1.17.0
+ tifffile==2025.5.10

```

```

(multimed) C:\Rafki Mulmed>

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

(multimed) C:\Rafki Mulmed>uv pip install moviepy
Using Python 3.10.18 environment at: multimed
Resolved 10 packages in 281ms
Installed 7 packages in 97ms
+ 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

```

The screenshot displays a Jupyter Notebook environment within a code editor. The top panel shows the Explorer and Code editors. The bottom panel shows the Terminal and Output windows.

Code Editor (Top):

```
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)}")
```

Terminal (Bottom):

```
sts, numba, lazy_loader, cffi, soundfile, scikit-learn, pooch, librosa
Successfully installed audiop-lts-0.2.2 audioread-3.0.1 certifi-2025.8.3 cffi-1.17.1 charset
_normalizer-3.4.3 decorator-5.2.1 idna-3.10 joblib-1.5.2 lazy_loader-0.4 librosa-0.11.0 llvml
ite-0.44.0 msgpack-1.1.1 numba-0.61.2 numpy-2.2.6 packaging-25.0 platformdirs-4.4.0 pooch-1.8
.2 pycparser-2.22 requests-2.32.5 scikit-learn-1.7.1 scipy-1.16.1 soundfile-0.13.1 soxr-0.5.0
.post1 standard-aifc-3.13.0 standard-chunk-3.13.0 standard-sunau-3.13.0 threadpoolctl-3.6.0 t
yping_extensions-4.15.0 urllib3-2.5.0

(multimed) C:\Rafki Mulmed>
```

Code Editor (Bottom):

```
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)}")
```


Output (Bottom):

Sine Wave (440 Hz)

The plot shows a sine wave with a frequency of 440 Hz. The x-axis represents time in seconds, and the y-axis represents amplitude, ranging from 0.25 to 1.00. The plot shows approximately 10 full cycles of the sine wave.


```
worksheet1.tex 2 rafki.ipynb x Readme.MD
Worksheet 1 > rafki.ipynb
+ Code + Markdown ...
▶
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')

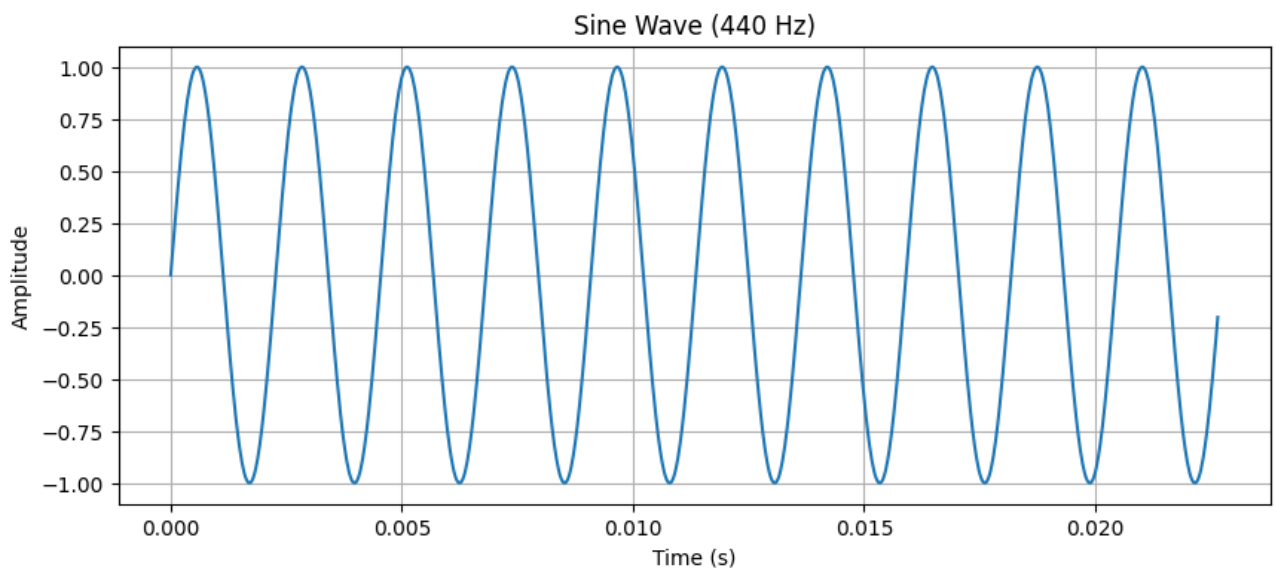
[13]
...
Test Image with RGB Stripes and White Circle
```



PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

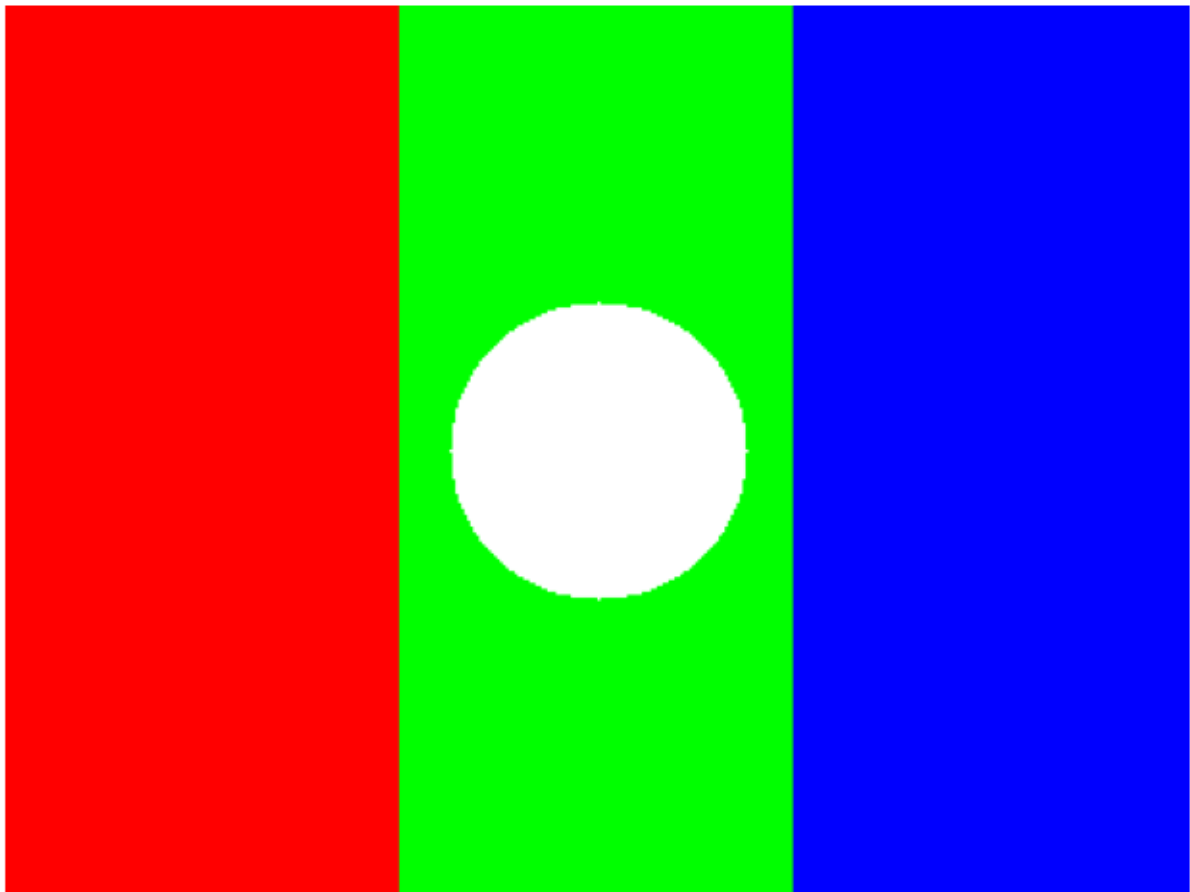
Microsoft Windows [Version 10.0.26100.4946]
(c) Microsoft Corporation. All rights reserved.

- Terminal/command prompt yang menunjukkan environment aktif
- Rafki Mulmed Output pertama _multimedia.py



- Rafki Mulmet Output kedua _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, penggunaan environment terpisah itu penting supaya proyek tetap stabil dan tidak terpengaruh oleh dependensi yang ada di proyek lain. Selain itu, dengan environment yang terisolasi, saya bisa lebih leluasa melakukan eksperimen tanpa takut merusak sistem atau proyek lain yang sedang berjalan. Ini juga saran dari teman yang sudah berpengalaman, dan menurut mereka ini cara yang paling efektif untuk menghindari masalah kompatibilitas.

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

Secara umum, menurut saya uv itu lebih modern dan lebih cepat dibandingkan venv, apalagi dalam hal setup awalnya. Sementara conda lebih lengkap karena sudah menyertakan package manager, jadi lebih mudah kalau membutuhkan library non-Python atau yang kompleks. Saya pribadi memilih uv karena lebih ringan dan cepat, dan juga karena sudah banyak teman yang merekomendasikannya, jadi saya merasa lebih nyaman menggunakannya.

3. Library mana yang paling sulit diinstall dan mengapa?

Sejauh ini, saya merasa semua library yang saya butuhkan bisa diinstall dengan lancar tanpa kendala berarti. Saya rasa karena saya sudah menggunakan environment yang bersih, jadi proses

instalasi jadi lebih mulus dan tidak ada konflik yang berarti. Jadi, sejauh ini tidak ada yang terlalu sulit diinstall.

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

Biasanya, kalau ada masalah dependency conflict, cara pertama yang saya coba adalah memastikan bahwa saya menggunakan versi yang tepat dari masing-masing library. Jika itu masih tidak berhasil, saya biasanya membuat environment baru agar lebih bersih, atau mencoba mengupdate salah satu library yang bermasalah. Biasanya itu cukup efektif untuk menyelesaikan masalah conflict.

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

Jelaskan fungsi dari masing-masing library yang berhasil Anda install! Librosa saya gunakan untuk pengolahan audio, seperti ekstraksi fitur atau analisis sinyal audio. Soundfile berguna untuk membaca dan menulis file audio dalam berbagai format. Scipy lebih banyak digunakan untuk komputasi ilmiah, terutama yang berkaitan dengan pemrosesan sinyal. OpenCV dan Pillow saya pakai untuk pengolahan citra, seperti manipulasi gambar dan pengolahan video. Scikit-image fokus pada analisis citra, sementara Matplotlib digunakan untuk visualisasi data dalam bentuk grafik. FFmpeg membantu dalam pemrosesan video, sedangkan MoviePy digunakan untuk editing video. NumPy dan Pandas saya gunakan untuk komputasi numerik dan manipulasi data, sementara Jupyter sangat berguna untuk eksperimen interaktif dan dokumentasi kode.

4.4 Troubleshooting

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

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

Solusi: *Saya menggunakan fitur Co-Pilot untuk mencari solusi, dan setelah menambahkan usepackagefont, masalah tersebut bisa teratasi.*

- **Masalah 2:** *[Saat mengikuti tutorial output1]*

Solusi: *Dengan bantuan Co-Pilot, saya langsung menemukan solusi dan perbaikan yang tepat, jadi masalah bisa segera diperbaiki.*

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 13: 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 ]

```

[rafki.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

Menurut saya, proses setup Python environment untuk proyek multimedia sangat penting untuk memastikan bahwa lingkungan kerja tetap stabil dan terpisah dari proyek lain. Dengan menggunakan environment terisolasi, kita bisa mencegah masalah kompatibilitas antar library dan memastikan bahwa proyek bisa dijalankan ulang di masa depan tanpa masalah. Untuk proyek multimedia berikutnya, saya akan lebih fokus pada manajemen dependensi dan memastikan semua library yang digunakan sudah sesuai versi yang dibutuhkan. Bagi mahasiswa lain yang akan melakukan hal serupa, saya sarankan untuk selalu membaca dokumentasi library dengan teliti dan jangan sungkan untuk mencari bantuan dari komunitas jika mengalami kesulitan. Hal ini akan sangat membantu untuk menghemat waktu dan menghindari masalah teknis yang tidak perlu.

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.