



Nama: **Rizki Alfariz Ramadhan (122140061)**
Environment untuk Multimedia

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

Mata Kuliah: **Sistem Teknologi Multimedia (IF25-40305)**

Tanggal: August 29, 2025

1 Instruksi Tugas

1.1 Persiapan

Sebelum memulai, pastikan Anda telah:

- Menginstall Python 3.8 atau lebih baru di sistem Anda. Versi python yang digunakan adalah 3.10.18
- Memilih salah satu tool manajemen environment: **conda**, **venv**, atau **uv**. Salah satu tool manajemen environment yang digunakan adalah **uv**.
- Membuka terminal/command prompt
- Menyiapkan dokumen L^AT_EX ini untuk dokumentasi

1.2 Bagian 1: Membuat Environment Python

1.2.1 Opsi 3: Menggunakan uv (Modern dan cepat)

```
1 # Install uv terlebih dahulu jika belum ada
2 pip install uv
3
4 # Jika menggunakan PowerShell (Windows)
5 powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
6
7 # Membuat environment baru
8 uv venv multimedia-uv
9
10 # Mengaktifkan environment (Windows)
11 multimedia-uv\Scripts\activate
12
13 # Downgrade versi Python
14 uv python install 3.10
15
16 # Verifikasi environment aktif (Windows PowerShell)
17 Get-Command python
```

Kode 1: Membuat environment dengan uv

Dokumentasikan di sini:

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

```

PS E:\kiki\mulmed> pip install uv
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: uv in c:\users\acerl\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kf
ra8p0\localcache\local-packages\python313\site-packages (0.8.13)
PS E:\kiki\mulmed> powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
Downloading uv 0.8.13 (x86_64-pc-windows-msvc)
Installing to C:\Users\acerl\local\bin
  uv.exe
  uvx.exe
  uvw.exe
everything's installed!
PS E:\kiki\mulmed> uv venv multimedia-uv
Using CPython 3.10.18
Creating virtual environment at: multimedia-uv
Activate with: multimedia-uv\Scripts\activate
PS E:\kiki\mulmed> multimedia-uv\Scripts\activate
(multimedia-uv) PS E:\kiki\mulmed> uv python install 3.10
(multimedia-uv) PS E:\kiki\mulmed> Get-Command Python

CommandType      Name
-----
Application      python.exe
Version          3.10.18...
Source           E:\kiki\mulmed\multimedia-uv\Scripts\p...

(multimedia-uv) PS E:\kiki\mulmed> |

```

Gambar 1: Output perintah verifikasi environment

1.3 Bagian 2: Instalasi Library Multimedia

Setelah environment aktif, install library-library berikut:

1.3.1 Library Audio Processing

```

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

```

Kode 2: Instalasi library audio

1.3.2 Library Image Processing

```

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

```

Kode 3: Instalasi library image

1.3.3 Library Video Processing

```

1 # Untuk pip (venv/uv):
2 pip install moviepy ffmpeg

```

Kode 4: Instalasi library video

1.3.4 Library General Purpose

```

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

```

Kode 5: Instalasi library umum

Dokumentasikan di sini:

- Perintah instalasi yang Anda gunakan:

- Audio : `uv pip install librosa soundfile scipy`
- Image: `uv pip install opencv-python pillow scikit-image matplotlib`
- Video: `uv pip install moviepy ffmpeg`
- General: `uv pip install numpy pandas jupyter`
- Screenshot proses instalasi atau output sukses

```
(multimedia-uv) PS E:\Viki\multimed> uv pip install librosa soundfile scipy
Using Python 3.10.15 environment at: multimedia-uv
Resolved 25 packages in 1m 55s
Prepared 25 packages in 0.00s
Installing wheels...
warning: Failed to hardlink files; falling back to full copy. This may lead to degraded performance.
If the cache and target directories are on different filesystems, hardlinking may not be supported.
If this is intentional, set 'export UV_LINK_MODE=copy' or use '--link-mode=copy' to suppress this warning.
Installed 25 packages in 0.00s
+ librosa==0.10.0
+ soundfile==0.12.1
+ scipy==1.12.0
```

(a) Instalasi library audio processing

```
(multimedia-uv) PS E:\Viki\multimed> uv pip install opencv-python pillow scikit-image matplotlib
Using Python 3.10.15 environment at: multimedia-uv
Resolved 18 packages in 1m 51s
Prepared 18 packages in 0.00s
Installing wheels...
warning: Failed to hardlink files; falling back to full copy. This may lead to degraded performance.
If the cache and target directories are on different filesystems, hardlinking may not be supported.
If this is intentional, set 'export UV_LINK_MODE=copy' or use '--link-mode=copy' to suppress this warning.
Installed 18 packages in 0.00s
+ contourpy==1.3.0
+ cycler==0.12.1
+ fonttools==4.53.0
+ imageio==2.35.0
+ kiwisolver==1.4.5
+ matplotlib==3.9.0
+ networkx==3.3
+ opencv-python==4.10.0.84
+ pillow==11.0.0
+ pygments==2.18.0
+ python-dateutil==2.9.0.post0
+ scikit-image==0.22.0
+ six==1.17.0
+ tifffile==2025.5.10
```

(b) Instalasi library image processing

```
(multimedia-uv) PS E:\Viki\multimed> uv pip install moviepy ffmpeg
Using Python 3.10.15 environment at: multimedia-uv
Resolved 11 packages in 39.32s
Prepared 7 packages in 1m 15s
Installing wheels...
warning: Failed to hardlink files; falling back to full copy. This may lead to degraded performance.
If the cache and target directories are on different filesystems, hardlinking may not be supported.
If this is intentional, set 'export UV_LINK_MODE=copy' or use '--link-mode=copy' to suppress this warning.
Installed 7 packages in 70ms
+ colorama==0.4.6
+ ffmpeg==1.4
+ imageio-ffmpeg==0.0.0
+ moviepy==2.2.1
+ proglog==0.1.12
+ python-dotenv==1.1.1
+ tqdm==4.67.1
```

(c) Instalasi library video processing

```
(multimedia-uv) PS E:\Viki\multimed> uv pip install numpy pandas jupyter
Using Python 3.10.15 environment at: multimedia-uv
Resolved 25 packages in 1m 55s
Prepared 25 packages in 0.00s
Installing wheels...
warning: Failed to hardlink files; falling back to full copy. This may lead to degraded performance.
If the cache and target directories are on different filesystems, hardlinking may not be supported.
If this is intentional, set 'export UV_LINK_MODE=copy' or use '--link-mode=copy' to suppress this warning.
Installed 25 packages in 0.00s
+ numpy==2.0.2
+ pandas==2.2.3
+ jupyter==1.0.0
+ jupyterlab==4.4.6
+ jupyterlab-pygments==0.3.0
+ jupyterlab-server==2.27.3
+ jupyterlab-widgets==3.0.15
+ kiwisolver==1.4.5
+ 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==3.4.2
+ notebook==7.4.5
+ notebook-shim==0.2.4
+ numba==0.61.2
+ numpy==2.0.2
+ opencv-python==4.12.0.88
+ overrides==7.7.0
+ packaging==25.0
+ pandas==2.3.2
+ pandocfilters==1.5.1
```

(d) Instalasi library general purpose

Gambar 2: Output instalasi library

- Daftar library yang berhasil diinstall dengan versinya:

anyio==4.10.0	h11==0.16.0	jupyterlab-server==2.27.3
argon2-cffi==25.1.0	httpcore==1.0.9	jupyterlab-widgets==3.0.15
argon2-cffi-bindings==25.1.0	httpx==0.28.1	kiwisolver==1.4.9
arrow==1.3.0	idna==3.10	lark==1.2.2
asttokens==3.0.0	imageio==2.37.0	lazy-loader==0.4
async-lru==2.0.5	imageio-ffmpeg==0.6.0	librosa==0.11.0
attrs==25.3.0	ipykernel==6.30.1	llvmlite==0.44.0
audioread==3.0.1	ipython==8.37.0	markupsafe==3.0.2
babel==2.17.0	ipywidgets==8.1.7	matplotlib==3.10.5
beautifulsoup4==4.13.5	isoduration==20.11.0	matplotlib-inline==0.1.7
bleach==6.2.0	jedi==0.19.2	mistune==3.1.3
certifi==2025.8.3	jinja2==3.1.6	moviepy==2.2.1
cffi==1.17.1	joblib==1.5.2	msgpack==1.1.1
charset-normalizer==3.4.3	json5==0.12.1	nbclient==0.10.2
colorama==0.4.6	jsonpointer==3.0.0	nbconvert==7.16.6
comm==0.2.3	jsonschema==4.25.1	nbformat==5.10.4
contourpy==1.3.2	jsonschema-specifications==2025.4.1	nest-asyncio==1.6.0
cycler==0.12.1	jupyter==1.1.1	networkx==3.4.2
debugpy==1.8.16	jupyter-client==8.6.3	notebook==7.4.5
decorator==5.2.1	jupyter-console==6.6.3	notebook-shim==0.2.4
defusedxml==0.7.1	jupyter-core==5.8.1	numba==0.61.2
exceptiongroup==1.3.0	jupyter-events==0.12.0	numpy==2.2.6
executing==2.2.0	jupyter-lsp==2.3.0	opencv-python==4.12.0.88
fastjsonschema==2.21.2	jupyter-server==2.17.0	overrides==7.7.0
ffmpeg==1.4	jupyter-server-terminals==0.5.3	packaging==25.0
fonttools==4.59.2	jupyterlab==4.4.6	pandas==2.3.2
fqdn==1.5.1	jupyterlab-pygments==0.3.0	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
pyparser==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==1.15.3
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
terminado==0.18.1

threadpoolctl==3.6.0
tifffile==2025.5.10
tinycss2==1.4.0
tomli==2.2.1
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

```

1.4 Bagian 3: Verifikasi Instalasi

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

```

1 import pkg_resources
2
3 requirements = [
4     "anyio==4.10.0",
5     "argon2-cffi==25.1.0",
6     "argon2-cffi-bindings==25.1.0",
7     "arrow==1.3.0",
8     "asttokens==3.0.0",
9     "async-lru==2.0.5",
10    "attrs==25.3.0",
11    "audioread==3.0.1",
12    "babel==2.17.0",
13    "beautifulsoup4==4.13.5",
14    "bleach==6.2.0",
15    "certifi==2025.8.3",
16    "cffi==1.17.1",
17    "charset-normalizer==3.4.3",
18    "colorama==0.4.6",
19    "comm==0.2.3",
20    "contourpy==1.3.2",
21    "cycler==0.12.1",
22    "debugpy==1.8.16",
23    "decorator==5.2.1",
24    "defusedxml==0.7.1",
25    "exceptiongroup==1.3.0",
26    "executing==2.2.0",
27    "fastjsonschema==2.21.2",
28    "ffmpeg==1.4",
29    "fonttools==4.59.2",
30    "fqdn==1.5.1",
31    "h11==0.16.0",
32    "httpcore==1.0.9",
33    "httpx==0.28.1",
34    "idna==3.10",
35    "imageio==2.37.0",
36    "imageio-ffmpeg==0.6.0",
37    "ipykernel==6.30.1",
38    "ipython==8.37.0",
39    "ipywidgets==8.1.7",
40    "isoduration==20.11.0",
41    "jedi==0.19.2",

```

```
42 "jinja2==3.1.6",
43 "joblib==1.5.2",
44 "json5==0.12.1",
45 "jsonpointer==3.0.0",
46 "jsonschema==4.25.1",
47 "jsonschema-specifications==2025.4.1",
48 "jupyter==1.1.1",
49 "jupyter-client==8.6.3",
50 "jupyter-console==6.6.3",
51 "jupyter-core==5.8.1",
52 "jupyter-events==0.12.0",
53 "jupyter-lsp==2.3.0",
54 "jupyter-server==2.17.0",
55 "jupyter-server-terminals==0.5.3",
56 "jupyterlab==4.4.6",
57 "jupyterlab-pygments==0.3.0",
58 "jupyterlab-server==2.27.3",
59 "jupyterlab-widgets==3.0.15",
60 "kiwisolver==1.4.9",
61 "lark==1.2.2",
62 "lazy-loader==0.4",
63 "librosa==0.11.0",
64 "llvmlite==0.44.0",
65 "markupsafe==3.0.2",
66 "matplotlib==3.10.5",
67 "matplotlib-inline==0.1.7",
68 "mistune==3.1.3",
69 "moviepy==2.2.1",
70 "msgpack==1.1.1",
71 "nbclient==0.10.2",
72 "nbconvert==7.16.6",
73 "nbformat==5.10.4",
74 "nest-asyncio==1.6.0",
75 "networkx==3.4.2",
76 "notebook==7.4.5",
77 "notebook-shim==0.2.4",
78 "numba==0.61.2",
79 "numpy==2.2.6",
80 "opencv-python==4.12.0.88",
81 "overrides==7.7.0",
82 "packaging==25.0",
83 "pandas==2.3.2",
84 "pandocfilters==1.5.1",
85 "parso==0.8.5",
86 "pillow==11.3.0",
87 "platformdirs==4.4.0",
88 "pooch==1.8.2",
89 "proglog==0.1.12",
90 "prometheus-client==0.22.1",
91 "prompt-toolkit==3.0.52",
92 "psutil==7.0.0",
93 "pure-eval==0.2.3",
94 "pycparser==2.22",
95 "pygments==2.19.2",
96 "pyparsing==3.2.3",
97 "python-dateutil==2.9.0.post0",
98 "python-dotenv==1.1.1",
99 "python-json-logger==3.3.0",
100 "pytz==2025.2",
101 "pywin32==311",
102 "pywinpty==3.0.0",
103 "pyyaml==6.0.2",
```

```

104 "pymzq==27.0.2",
105 "referencing==0.36.2",
106 "requests==2.32.5",
107 "rfc3339-validator==0.1.4",
108 "rfc3986-validator==0.1.1",
109 "rfc3987-syntax==1.1.0",
110 "rpds-py==0.27.1",
111 "scikit-image==0.25.2",
112 "scikit-learn==1.7.1",
113 "scipy==1.15.3",
114 "send2trash==1.8.3",
115 "setuptools==80.9.0",
116 "six==1.17.0",
117 "sniffio==1.3.1",
118 "soundfile==0.13.1",
119 "soupsieve==2.8",
120 "soxr==0.5.0.post1",
121 "stack-data==0.6.3",
122 "terminado==0.18.1",
123 "threadpoolctl==3.6.0",
124 "tifffile==2025.5.10",
125 "tinycss2==1.4.0",
126 "tomli==2.2.1",
127 "tornado==6.5.2",
128 "tqdm==4.67.1",
129 "traitlets==5.14.3",
130 "types-python-dateutil==2.9.0.20250822",
131 "typing-extensions==4.15.0",
132 "tzdata==2025.2",
133 "uri-template==1.3.0",
134 "urllib3==2.5.0",
135 "wcwidth==0.2.13",
136 "webcolors==24.11.1",
137 "webencodings==0.5.1",
138 "websocket-client==1.8.0",
139 "widgetsnbextension==4.0.14",
140 ]
141
142 def check_requirements(reqs):
143     installed_packages = {pkg.key: pkg.version for pkg in pkg_resources.working_set}
144
145     print(f"{'Package':30} {'Required':15} {'Installed':15} Status")
146     print("="*80)
147
148     for req in reqs:
149         try:
150             req_pkg = pkg_resources.Requirement.parse(req)
151             name = req_pkg.key
152             required_version = str(req_pkg.specifier) if req_pkg.specifier else "Any"
153         except Exception as e:
154             print(f"{'req':30} {'-':15} {'-':15} Error parsing")
155             continue
156
157         if name in installed_packages:
158             installed_version = installed_packages[name]
159             if req_pkg.specifier and installed_version not in req_pkg.specifier:
160                 status = "Version mismatch"
161             else:
162                 status = "OK"
163         else:
164             installed_version = "-"
165             status = "Missing"

```

```

166
167     print(f"{name:30} {required_version:15} {installed_version:15} {status}")
168
169 if __name__ == "__main__":
170     check_requirements(requirements)

```

Kode 6: Kode pengujian library

Jalankan script dan dokumentasikan hasilnya:

```

[MultiMedia-00] PS E:\MultiMedia> python E:\MultiMedia\Scripts\testPackage.py
E:\MultiMedia\Scripts\testPackage.py:1: UserWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for removal as early as 2025-11-30. Refrain
from using this package or pin to setuptools<81.
  import pkg_resources
Report pkg_resources
Package Required Installed Status
-----
anyio ==4.10.0 4.10.0 OK
argon2-cffi ==25.1.0 25.1.0 OK
argon2-cffi-bindings ==25.1.0 25.1.0 OK
arrow ==1.3.0 1.3.0 OK
asttokens ==3.0.0 3.0.0 OK
async-lru ==2.0.5 2.0.5 OK
attrs ==25.3.0 25.3.0 OK
audioread ==3.0.4 3.0.4 OK
babel ==2.17.0 2.17.0 OK
beautifulsoup4 ==4.13.5 4.13.5 OK
bleach ==6.2.0 6.2.0 OK
certifi ==2025.8.3 2025.8.3 OK
cffi ==1.17.1 1.17.1 OK
charset-normalizer ==3.4.3 3.4.3 OK
colorama ==0.4.6 0.4.6 OK
comm ==0.2.2 0.2.2 OK
contourpy ==1.3.2 1.3.2 OK
cython ==0.12.1 0.12.1 OK
debugpy ==1.8.16 1.8.16 OK
decorator ==5.2.1 5.2.1 OK
defusedxml ==0.7.1 0.7.1 OK
exceptiongroup ==1.3.0 1.3.0 OK
executing ==2.2.0 2.2.0 OK
fastjsonschema ==2.21.2 2.21.2 OK
ffmpeg ==1.4 1.4 OK
fonttools ==4.59.2 4.59.2 OK
fpdf ==1.5.1 1.5.1 OK
h11 ==0.16.0 0.16.0 OK
httpcore ==1.0.9 1.0.9 OK
httpx ==0.28.1 0.28.1 OK
idna ==3.10 3.10 OK
imageio ==2.37.0 2.37.0 OK
imageio-ffmpeg ==0.6.0 0.6.0 OK
ipykernel ==6.30.1 6.30.1 OK
ipython ==8.37.0 8.37.0 OK
ipywidgets ==8.1.7 8.1.7 OK
isoduration ==0.11.0 0.11.0 OK
jedi ==0.19.2 0.19.2 OK
Jinja2 ==3.1.6 3.1.6 OK
jlab ==1.5.2 1.5.2 OK
json5 ==0.12.1 0.12.1 OK
jsonpointer ==3.0.0 3.0.0 OK
jsonschema ==4.25.1 4.25.1 OK
jsonschema-specifications ==2025.4.1 2025.4.1 OK
jupyter ==1.1.1 1.1.1 OK
jupyter-client ==8.6.3 8.6.3 OK
jupyter-console ==6.6.3 6.6.3 OK
jupyter-core ==5.1.1 5.1.1 OK
jupyter-events ==0.12.0 0.12.0 OK
jupyter-lsp ==2.3.0 2.3.0 OK
jupyter-server ==2.17.0 2.17.0 OK
jupyter-server-terminals ==0.5.3 0.5.3 OK
jupyterlab ==4.4.6 4.4.6 OK
jupyterlab-pygments ==0.3.0 0.3.0 OK
jupyterlab-server ==2.27.3 2.27.3 OK
jupyterlab-widgets ==3.0.15 3.0.15 OK
kiwisolver ==1.4.9 1.4.9 OK
latex ==1.2.2 1.2.2 OK
lazy-loader ==0.4 0.4 OK
librosa ==0.11.0 0.11.0 OK

```

Gambar 3: Output pengujian library

1.5 Bagian 4: Simple Test dengan Sample Code

Buat dan jalankan contoh sederhana untuk setiap kategori multimedia:

1.5.1 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 7: Test audio processing sederhana

1.5.2 Test Image Processing

```

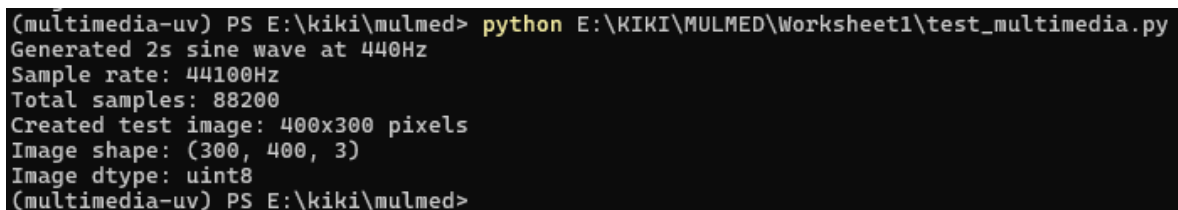
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from PIL import Image
4
5 # Create a simple test image
6 width, height = 400, 300
7 image = np.zeros((height, width, 3), dtype=np.uint8)
8
9 # Add some patterns
10 image[:, :width//3, 0] = 255 # Red section
11 image[:, width//3:2*width//3, 1] = 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 8: Test image processing sederhana

Dokumentasikan hasil eksekusi:

- Screenshot output dari kedua script di atas



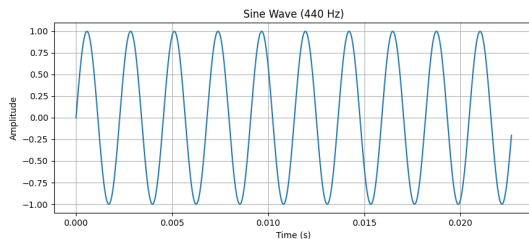
```

(multimedia-uv) PS E:\kiki\mulmed> python E:\KIKI\MULMED\Worksheet1\test_multimedia.py
Generated 2s sine wave at 440Hz
Sample rate: 44100Hz
Total samples: 88200
Created test image: 400x300 pixels
Image shape: (300, 400, 3)
Image dtype: uint8
(multimedia-uv) PS E:\kiki\mulmed>

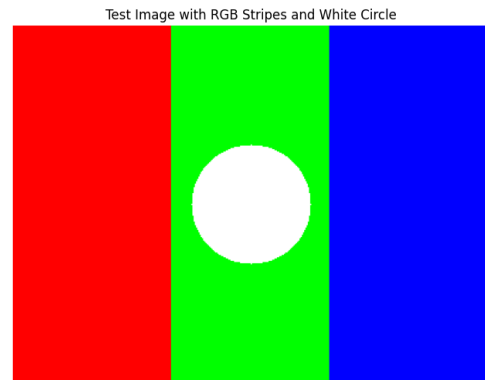
```

Gambar 4: Output pengujian library

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



(a) Visualisasi sine wave 440 Hz



(b) Visualisasi test image

Gambar 5: Gambar yang dihasilkan dari script test multimedia

2 Bagian Laporan

2.1 Output Verifikasi Instalasi

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

```
1 Generated 2s sine wave at 440Hz
2 Sample rate: 44100Hz
3 Total samples: 88200
4 Created test image: 400x300 pixels
5 Image shape: (300, 400, 3)
6 Image dtype: uint8
```

Kode 9: Output verifikasi instalasi

2.2 Screenshot Hasil Test

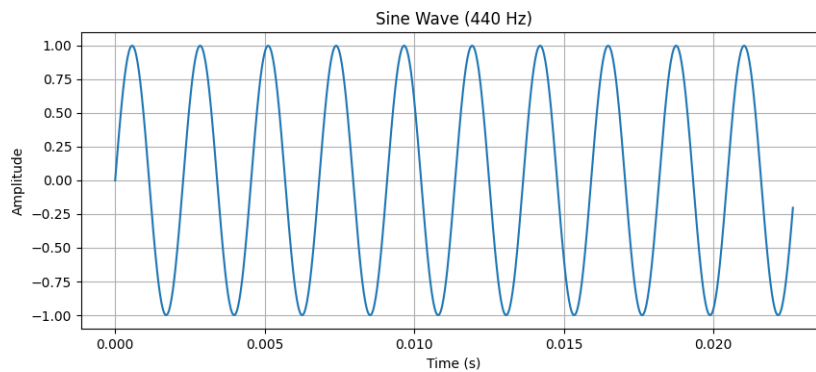
Sisipkan screenshot atau gambar hasil dari:

- Terminal/command prompt yang menunjukkan environment aktif

```
(multimedia-uv) PS E:\kiki\mulmed> python E:\KIKI\MULMED\Worksheet1\test_multimedia.py
Generated 2s sine wave at 440Hz
Sample rate: 44100Hz
Total samples: 88200
Created test image: 400x300 pixels
Image shape: (300, 400, 3)
Image dtype: uint8
(multimedia-uv) PS E:\kiki\mulmed>
```

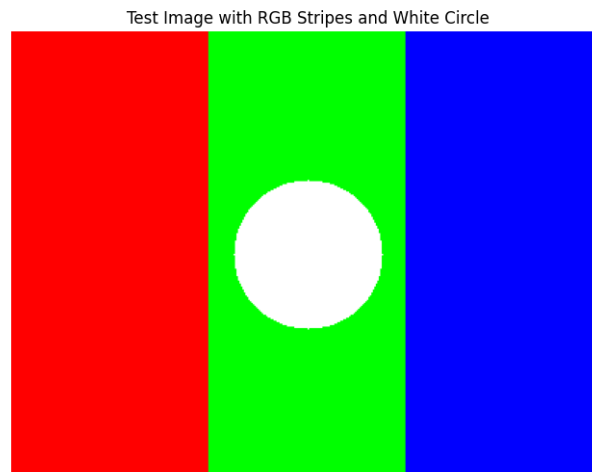
Gambar 6: Output test multimedia di terminal

- Output dari script test audio (sine wave plot)



Gambar 7: Output test audio dengan sine wave 440 Hz

- Output dari script test image (RGB stripes dengan circle)



Gambar 8: Output test image dengan RGB stripes dan circle

2.3 Analisis dan Refleksi

Jawab pertanyaan berikut:

1. Mengapa penting menggunakan environment terpisah untuk project multimedia?

Dengan menggunakan environment terpisah, kita dapat mengelola dependensi library yang spesifik untuk project multimedia tanpa mengganggu instalasi global Python atau project lain. Hal ini membantu mencegah konflik versi library dan memastikan bahwa project berjalan dengan konfigurasi yang konsisten.

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

Conda adalah manajer environment yang juga mengelola paket dan dependensi, sesuai mengelola dependensi kompleks di berbagai bahasa pemrograman. Venv adalah tool bawaan Python yang sederhana untuk membuat environment terisolasi dan sesuai untuk proyek-proyek sederhana yang hanya menggunakan Python, sedangkan uv adalah tool modern yang cepat dan mudah dan

dirancang untuk mengelola lingkungan dan menginstal paket dengan efisiensi tinggi. Alasan memilih uv adalah karena kemudahan penggunaan dan kecepatan dalam membuat environment.

3. Library mana yang paling sulit diinstall dan mengapa?

Sejauh ini, tidak ada library yang sulit diinstall. Semua library berhasil diinstall tanpa masalah.

4. Bagaimana cara mengatasi masalah dependency conflict jika terjadi?

Untuk mengatasi masalah dependency conflict, saya mencoba beberapa langkah berikut:

- *Mencari tahu library mana yang menyebabkan konflik.*
- *Menghapus library yang bermasalah dan menginstal ulang versi yang kompatibel.*
- *Memeriksa dokumentasi resmi library untuk mengetahui versi yang kompatibel.*
- *Menggunakan virtual environment terpisah untuk setiap project.*
- *Mencari solusi di forum atau dokumentasi resmi library terkait.*
- *Jika tetap tidak berhasil bertanya ke asisten virtual*

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

- **Librosa** digunakan untuk memuat, menganalisis, dan mengekstrak audio (MFCC, Spectrogram mel, Kroma).
- **soundfile** digunakan untuk membaca dan menulis berbagai berkas audio seperti WAV, FLAC, dll.
- **scipy** digunakan untuk melakukan komputasi ilmiah dan matematika, seperti optimasi, aljabar linear, interpolasi, pemrosesan gambar dan sinyal, serta statistik.
- **opencv-python** merupakan library yang dirancang untuk melakukan pengeditan dan tugas-tugas computer vision.
- **pillow** digunakan untuk memanipulasi gambar dan pemrosesan gambar dasar.
- **scikit-image** adalah kumpulan algoritma yang digunakan untuk memproses gambar dan computer vision.
- **matplotlib** digunakan untuk membuat visualisasi data secara menarik dan informatif
- **moviepy** digunakan untuk mengedit dan memproses video.
- **ffmpeg** merupakan kerangka kerja multimedia yang digunakan untuk menangani video, audio, dan berkas aliran multimedia lainnya.
- **numpy** digunakan untuk melakukan perhitungan saintifik seperti matriks, aljabar, dan lainnya.
- **pandas** digunakan untuk menganalisis, membersihkan, eksplorasi, dan manipulasi data
- **jupyter** digunakan sebagai lingkungan interaktif yang memungkinkan pengguna untuk menulis dan menjalankan kode serta menyajikan hasilnya.

2.4 Troubleshooting

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

- **Masalah 1:** Permasalahan instalasi MikTex dan Strawberry Perl di Windows, karena terbalik dalam proses instalasi.

Solusi: menginstal ulang MikTex dan Strawberry Perl dengan urutan yang benar, yaitu menginstal Strawberry Perl terlebih dahulu, kemudian MikTex.

- **Masalah 2:** Permasalahan dalam interpreter versi Python di VSCode yang tidak sesuai dengan versi environment yang dibuat.

Solusi: Bertanya ke teman (Fathan) untuk mengatasi masalah tersebut, kemudian mengubah interpreter di VSCode ke versi Python yang sesuai dengan environment yang telah dibuat.

3 Export Environment untuk Reproduksi

Sebagai langkah terakhir, export environment Anda agar dapat direproduksi:

3.1 Untuk venv/uv

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

Kode 10: Export pip requirements

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

```
1 anyio==4.10.0
2 argon2-cffi==25.1.0
3 argon2-cffi-bindings==25.1.0
4 arrow==1.3.0
5 asttokens==3.0.0
6 async-lru==2.0.5
7 attrs==25.3.0
8 audioread==3.0.1
9 babel==2.17.0
10 beautifulsoup4==4.13.5
11 bleach==6.2.0
12 certifi==2025.8.3
13 cffi==1.17.1
14 charset-normalizer==3.4.3
15 colorama==0.4.6
16 comm==0.2.3
17 contourpy==1.3.2
18 cycler==0.12.1
19 debugpy==1.8.16
20 decorator==5.2.1
21 defusedxml==0.7.1
22 exceptiongroup==1.3.0
23 executing==2.2.0
24 fastjsonschema==2.21.2
25 ffmpeg==1.4
26 fonttools==4.59.2
27 fqdn==1.5.1
28 h11==0.16.0
29 httpcore==1.0.9
30 httpx==0.28.1
31 idna==3.10
32 imageio==2.37.0
33 imageio-ffmpeg==0.6.0
34 ipykernel==6.30.1
35 ipython==8.37.0
36 ipywidgets==8.1.7
37 isoduration==20.11.0
38 jedi==0.19.2
39 jinja2==3.1.6
40 joblib==1.5.2
41 json5==0.12.1
42 jsonpointer==3.0.0
43 jsonschema==4.25.1
```

```
44 jsonschema-specifications==2025.4.1
45 jupyter==1.1.1
46 jupyter-client==8.6.3
47 jupyter-console==6.6.3
48 jupyter-core==5.8.1
49 jupyter-events==0.12.0
50 jupyter-lsp==2.3.0
51 jupyter-server==2.17.0
52 jupyter-server-terminals==0.5.3
53 jupyterlab==4.4.6
54 jupyterlab-pygments==0.3.0
55 jupyterlab-server==2.27.3
56 jupyterlab-widgets==3.0.15
57 kiwisolver==1.4.9
58 lark==1.2.2
59 lazy-loader==0.4
60 librosa==0.11.0
61 llvmlite==0.44.0
62 markupsafe==3.0.2
63 matplotlib==3.10.5
64 matplotlib-inline==0.1.7
65 mistune==3.1.3
66 moviepy==2.2.1
67 msgpack==1.1.1
68 nbclient==0.10.2
69 nbconvert==7.16.6
70 nbformat==5.10.4
71 nest-asyncio==1.6.0
72 networkx==3.4.2
73 notebook==7.4.5
74 notebook-shim==0.2.4
75 numba==0.61.2
76 numpy==2.2.6
77 opencv-python==4.12.0.88
78 overrides==7.7.0
79 packaging==25.0
80 pandas==2.3.2
81 pandocfilters==1.5.1
82 parso==0.8.5
83 pillow==11.3.0
84 platformdirs==4.4.0
85 pooch==1.8.2
86 proglog==0.1.12
87 prometheus-client==0.22.1
88 prompt-toolkit==3.0.52
89 psutil==7.0.0
90 pure-eval==0.2.3
91 pycparser==2.22
92 pygments==2.19.2
93 pyparsing==3.2.3
94 python-dateutil==2.9.0.post0
95 python-dotenv==1.1.1
96 python-json-logger==3.3.0
97 pytz==2025.2
98 pywin32==311
99 pywinpty==3.0.0
100 pyyaml==6.0.2
101 pyzmq==27.0.2
102 referencing==0.36.2
103 requests==2.32.5
104 rfc3339-validator==0.1.4
105 rfc3986-validator==0.1.1
```

```
106 rfc3987-syntax==1.1.0
107 rpds-py==0.27.1
108 scikit-image==0.25.2
109 scikit-learn==1.7.1
110 scipy==1.15.3
111 send2trash==1.8.3
112 setuptools==80.9.0
113 six==1.17.0
114 sniffio==1.3.1
115 soundfile==0.13.1
116 soupsieve==2.8
117 soxr==0.5.0.post1
118 stack-data==0.6.3
119 terminado==0.18.1
120 threadpoolctl==3.6.0
121 tifffile==2025.5.10
122 tinycss2==1.4.0
123 tomli==2.2.1
124 tornado==6.5.2
125 tqdm==4.67.1
126 traitlets==5.14.3
127 types-python-dateutil==2.9.0.20250822
128 typing-extensions==4.15.0
129 tzdata==2025.2
130 uri-template==1.3.0
131 urllib3==2.5.0
132 wcwidth==0.2.13
133 webcolors==24.11.1
134 webencodings==0.5.1
135 websocket-client==1.8.0
136 widgetsnbextension==4.0.14
```

Kode 11: Environment/Requirements file

4 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

Pemisahan dependensi environment perlu dilakukan untuk menghindari konflik antar library serta memastikan stabilitas sistem yang digunakan. Dengan cara ini, setiap proyek dapat berjalan pada lingkungannya masing-masing tanpa saling memengaruhi, sehingga kesalahan akibat perbedaan versi library dapat diminimalisasi. Selain itu, pemisahan environment juga membantu dalam mengoptimalkan penggunaan sumber daya, karena hanya library yang benar-benar dibutuhkan saja yang diinstal. Persiapan ke depannya adalah selalu membuat environment baru untuk setiap proyek yang berbeda, kemudian menyesuaikan instalasi library sesuai kebutuhan. Hal ini tidak hanya memudahkan dalam proses pengembangan, tetapi juga mempermudah proses debugging ketika terjadi kesalahan. Saran untuk mahasiswa yang melakukan hal serupa adalah membiasakan diri untuk memisahkan dependensi environment, menggunakan manajer environment yang sesuai, serta selalu membaca dokumentasi resmi agar memahami setiap alur dalam proses instalasi. Dengan demikian, pengelolaan environment akan menjadi lebih terstruktur, efisien, dan mendukung keberhasilan proyek.

5 Referensi

- [Astral UV Documentation](#)
- [ChatGPT](#)

6 Lampiran

- [GitHub](#)