

# IMPLEMENTASI TEKNIK ROTASI GAMBAR MENGUNAKAN METODE AFFINE TRANSFORMATION DENGAN PYTHON DAN OPENCV

Laurah Shabrillah<sup>1</sup>, Rifal Fatahillah<sup>2</sup>, Romadhon Nanda Prastyo<sup>3</sup>, Sarah Azqia<sup>4</sup>, Perani Rosyani<sup>5</sup>

<sup>1</sup>Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Pamulang, Tangerang Selatan Banten, Indonesia

E:mail:<sup>1</sup>[laurahshabrillah218@gmail.com](mailto:laurahshabrillah218@gmail.com),<sup>2</sup>[rifalfatahillah@gmail.com](mailto:rifalfatahillah@gmail.com),<sup>3</sup>[romadhonnandaprastyo@gmail.com](mailto:romadhonnandaprastyo@gmail.com),<sup>4</sup>[sarahazqia08@gmail.com](mailto:sarahazqia08@gmail.com),<sup>5</sup>[dosen00837@unpam.ac.id](mailto:dosen00837@unpam.ac.id)

**Abstrak-** Rotasi gambar merupakan salah satu teknik fundamental dalam pengolahan citra digital yang memiliki berbagai aplikasi dalam bidang computer vision, pengolahan gambar medis, dan sistem keamanan visual. Penelitian ini berfokus pada implementasi teknik rotasi gambar menggunakan metode Affine Transformation dengan memanfaatkan bahasa pemrograman Python dan Library OpenCV. Tujuan utama dari penelitian ini adalah untuk menganalisis efektivitas dan efisiensi metode Affine Transformation dalam melakukan rotasi gambar, serta mengevaluasi kualitas hasil rotasi yang dihasilkan.

**Kata Kunci:** Rotasi Gambar, Affine Transformation, Python, OpenCV

**Abstract**—Image rotation is a fundamental technique in digital image processing with various applications in computer vision, medical image processing, and visual security systems. This research focuses on the implementation of image rotation techniques using the Affine Transformation method, utilizing the Python programming language and the OpenCV library. The primary objective of this study is to analyze the effectiveness and efficiency of the Affine Transformation method in performing image rotation and to evaluate the quality of the resulting rotated images.

**Keywords:** Image Rotation, Affine Transformation, Python, OpenCV.

## 1.PENDAHULUAN

Aplikasi menurut Dhanta dikutip dari Azhar (2019) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel. Aplikasi berasal dari kata application yang artinya penerapan lamaran penggunaan. Aplikasi adalah alat yang digunakan untuk membantu pekerjaan manusia menjadi lebih efisien dan efektif. Hal ini didukung dengan perkembangan aplikasi yang berkembang sangat pesat dari waktu ke waktu. Salah satu aplikasi yang membantu manusia adalah aplikasi rotasi gambar. Aplikasi rotasi gambar adalah aplikasi dalam bidang computer vision, pengolahan gambar medis, dan sistem keamanan visual. Dengan adanya aplikasi ini dapat mempermudah pengguna dalam melakukan kegiatan yang berkaitan dengan gambar. Aplikasi rotasi gambar ini menggunakan metode Affine Transformation. Metode Affine Transformation dapat melakukan rotasi gambar dengan tingkat presisi yang terukur, khususnya dalam upaya untuk mempertahankan kualitas dan detail gambar kemudian rotasi gambar. Implementasi menggunakan Python dan OpenCV telah berhasil memperlihatkan bahwa komputasi dapat dilakukan dengan baik, waktu pemrosesan relatif cepat termasuk untuk gambar yang memiliki resolusi tinggi. Kami mengembangkan metode Affine Transformation untuk meningkatkan akurasi dan kecepatan pengolahan gambar agar relevan untuk memberikan respons cepat dan kinerja yang handal.

## 2.METODE PENELITIAN

### 2.1 Metode Affine Transformation

Metode Affine Transformation adalah teknik dalam pengolahan citra digital yang digunakan untuk melakukan transformasi geometris pada gambar. Affine Transformation mencakup beberapa operasi dasar seperti translasi (pemindahan), skala (perubahan ukuran), rotasi (perputaran), dan shear (penggeseran), yang dapat diterapkan secara bersamaan atau terpisah. Dalam konteks

rotasi gambar, Affine Transformation menggunakan matriks transformasi untuk memutar gambar sekitar titik asal atau titik tertentu. Transformasi ini mempertahankan garis lurus dan proporsi paralel dari objek dalam gambar, sehingga hasil rotasi tidak mengubah bentuk objek, hanya orientasinya.

## 2.2 Implementasi Affine Transformation

**Tahap implementasi sistem merupakan proses penempatan sistem agar siap untuk dioperasikan, yang mencakup kegiatan menulis kode program (Perani Rosyani, 2021).** Dalam implementasinya menggunakan Python dan OpenCV, Affine Transformation dapat dilakukan dengan langkah-langkah berikut:

- a) Membuat matriks rotasi: Menggunakan fungsi `cv2.getRotationMatrix2D(center, angle, scale)` untuk mendapatkan matriks rotasi berdasarkan titik pusat rotasi, sudut rotasi, dan skala.
- b) Menerapkan transformasi: Menggunakan fungsi `cv2.warpAffine(src, M, dsize)` untuk menerapkan matriks transformasi M pada gambar sumber src dengan ukuran keluaran dsize.

Dalam Implementasi Affine Transformation telah terbukti efektif dalam aplikasi pengolahan gambar dan untuk mengevaluasi efektivitas dan efisiensi rotasi gambar, memastikan kualitas dan detail gambar tetap terjaga setelah rotasi, serta menganalisis distorsi dan artefak yang mungkin muncul selama proses rotasi.

## 3. ANALISA DAN PEMBAHASAAN

Bagian ini berisi analisis serta pembahasan dari topik penelitian, yang dapat dikembangkan setelah metodologi penelitian dibahas. Bagian ini akan memberikan penjelasan mengenai aplikasi rotasi gambar.

### 3.1 Fleksibilitas Python

**Program komputer dirancang untuk memudahkan manusia dalam menyelesaikan berbagai topik dengan memberikan jawaban atau saran setara dengan yang diberikan oleh pakar manusia (Perani Rosyani, 2023).** Python, sebagai bahasa pemrograman serbaguna, digunakan untuk berbagai keperluan seperti pengembangan aplikasi web, aplikasi desktop, Internet of Things (IoT), dan berbagai aplikasi lainnya. Python juga memiliki kemampuan integrasi dengan sistem basis data dan mampu membaca serta mengubah record, sehingga sering digunakan untuk prototyping atau pengembangan perangkat lunak secara cepat dan reliabel. Fleksibilitas dan kemudahan penggunaan Python membuatnya menjadi pilihan utama bagi banyak pengembang dalam mengimplementasikan berbagai aplikasi, termasuk aplikasi pengolahan citra digital.

### 3.2 Implementasi OpenCV

**Implementasi merupakan tahap yang bertujuan untuk menerapkan desain yang telah dirancang terhadap sistem, sehingga sistem dapat mencapai tujuan yang diinginkan oleh pengguna serta memberikan masukan dan melihat perkembangan desain yang telah dibangun (Perani Rosyani, 2021).** OpenCV (Open Computer Vision) adalah sebuah library API (Application Programming Interface) yang sangat populer dan luas digunakan dalam bidang pengolahan citra komputer. Salah satu keunggulan utama OpenCV adalah kebebasan penggunaannya untuk tujuan komersial maupun akademis. Algoritma-algoritma yang disertakan dalam OpenCV didesain agar bebas dari paten, sehingga pengguna dapat menggunakan dan mengembangkan aplikasi tanpa khawatir akan masalah lisensi atau pembatasan hukum. Hal ini menjadikan OpenCV sebagai pilihan utama bagi para pengembang yang ingin memanfaatkan teknologi pengolahan gambar tanpa terkendala dalam hal regulasi.

Pada dasarnya, OpenCV menyediakan berbagai algoritma dan fungsi yang dapat digunakan untuk mengolah gambar dan video secara efisien. Dalam konteks penelitian ini, OpenCV digunakan untuk menerapkan teknik rotasi gambar menggunakan metode Affine Transformation.

### 3.3 Implementasi Aplikasi Rotasi Gambar

Proses implementasi menggunakan metode Affine Transformation terdiri dari beberapa tahap . Setiap tahap memiliki tujuan khusus dan metode yang digunakan untuk mencapai hasil akhir yang diinginkan.

### 3.3.1 Langkah MengInstall OpenCV

Langkah-langkah berikut akan memandu Anda dalam menginstal OpenCV di lingkungan pengembangan Python. Pastikan Anda memiliki Python dan PIP (Python Package Manager) terinstal sebelum memulai.

a) Sebelum menginstal OpenCV, pastikan Python telah terinstal dengan benar di sistem Anda. Anda dapat memeriksa versi Python dengan menjalankan perintah berikut di terminal atau command prompt:

```
bash Copy code  
  
python --version
```

Pastikan juga pip terbaru telah terinstal dengan menjalankan:

```
bash Copy code  
  
pip install --upgrade pip
```

b) Instalasi OpenCV Menggunakan pip

OpenCV dapat diinstal dengan mudah menggunakan pip, yang merupakan package manager resmi untuk Python. Buka terminal atau command prompt dan jalankan perintah berikut:

```
bash Copy code  
  
pip install opencv-python
```

Perintah ini akan mengunduh dan menginstal paket OpenCV beserta dependensinya. Jika Anda memerlukan modul tambahan seperti operasi berbasis GPU, Anda dapat menginstalnya dengan:

```
bash Copy code  
  
pip install opencv-python-headless
```

c) Verifikasi Instalasi

Setelah instalasi selesai, pastikan OpenCV telah terinstal dengan benar. Buka Python shell atau buat skrip Python baru, lalu import modul cv2 (modul OpenCV):

```
python Copy code
```

```
import cv2  
print(cv2.__version__)
```

### 3.3.2 Langkah Membuat aplikasi Image Rotation Langkah

Berikut adalah langkah-langkah yang dapat Anda ikuti untuk menggabungkan keduanya:

- a) Pastikan untuk mengimpor semua library yang dibutuhkan baik untuk manipulasi gambar dengan OpenCV maupun untuk membuat antarmuka GUI dengan Tkinter.
- b) Definisikan kelas `ImageRotatorApp` untuk membuat antarmuka GUI dengan label gambar, slider untuk mengatur sudut rotasi, tombol untuk memuat gambar, dan tombol untuk menyimpan gambar yang sudah dirotasi.
- c) Implementasikan metode `load\_image` untuk memuat gambar dari sistem file dan menampilkannya di antarmuka menggunakan `ImageTk`.
- d) Buat metode `rotate\_image` untuk memutar gambar sesuai dengan sudut yang dipilih dari slider, menggunakan fungsi rotasi dari Pillow (PIL).
- f) Tambahkan metode `save\_rotated\_image` untuk menyimpan gambar yang sudah dirotasi ke lokasi yang dipilih oleh pengguna.
- g) Terakhir, definisikan fungsi `main` untuk membuat instance dari `tk.Tk()` (jendela utama Tkinter) dan instance dari `ImageRotatorApp`, serta memulai loop utama `root.mainloop()`.

Berikut adalah kode lengkap yang telah digabungkan:

```
import tkinter as tk  
from tkinter import filedialog  
from PIL import Image, ImageTk  
import numpy as np  
import cv2 as cv  
  
class ImageRotatorApp:  
    def __init__(self, master):  
        self.master = master  
        self.master.title(" Image Rotation")  
  
        # Buat label untuk menampilkan gambar  
        self.image_label = tk.Label(self.master)  
        self.image_label.pack()  
  
        # Buat slider untuk memasukkan sudut rotasi  
        self.rotation_slider = tk.Scale(self.master, from_=0, to=360, orient=tk.HORIZONTAL, label="Rotation  
Angle", command=self.rotate_image)  
        self.rotation_slider.pack()  
  
        # Tombol untuk memuat gambar  
        self.load_button = tk.Button(self.master, text="Load Image", command=self.load_image)  
        self.load_button.pack()
```

```
# Tombol untuk menyimpan gambar yang sudah dirotasi
self.save_button = tk.Button(self.master, text="Save Rotated Image", command=self.save_rotated_image)
self.save_button.pack()

# Inisialisasi variabel
self.image_path = None
self.original_image = None
self.rotated_image = None
self.rotation_angle = 0

def load_image(self):
# Memilih gambar dari sistem file
self.image_path = filedialog.askopenfilename(filetypes=[("Image files", ".jpg;.png;*.jpeg")])

# Memuat gambar asli menggunakan PIL (Pillow)
self.original_image = Image.open(self.image_path)

# Memuat gambar juga menggunakan OpenCV untuk tujuan demonstrasi
img_cv = cv.imread(self.image_path, cv.IMREAD_GRAYSCALE)
rows, cols = img_cv.shape
M = cv.getRotationMatrix2D(((cols-1)/2.0, (rows-1)/2.0), 90, 1)
dst_cv = cv.warpAffine(img_cv, M, (cols, rows))

# Konversi gambar hasil rotasi OpenCV ke format PIL untuk tampilan
pil_rotated_image = Image.fromarray(dst_cv)
self.display_image(pil_rotated_image)

def display_image(self, image):
# Mengonversi gambar PIL ke format yang dapat ditampilkan oleh Tkinter
image = ImageTk.PhotoImage(image)

# Menampilkan gambar di label
self.image_label.configure(image=image)
self.image_label.image = image

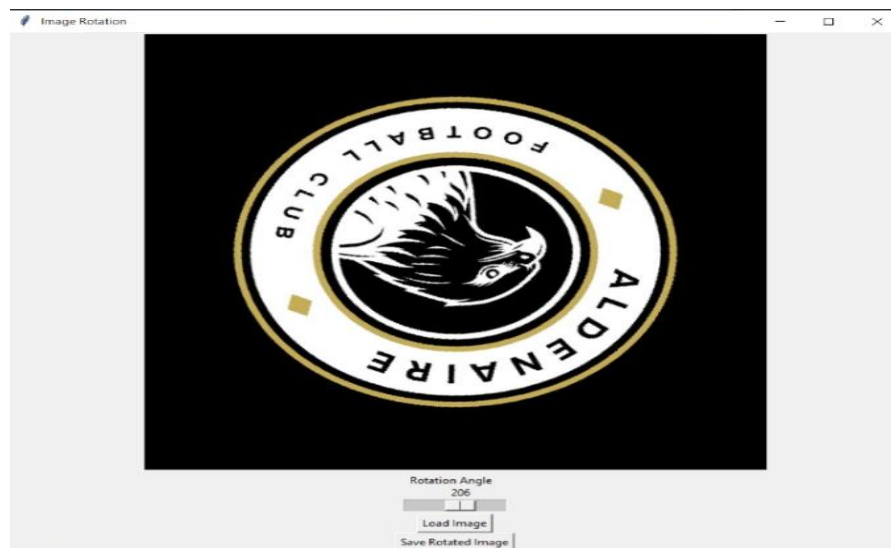
def rotate_image(self, angle):
# Memutar gambar sesuai dengan sudut rotasi yang dipilih
self.rotation_angle = int(angle)
self.rotated_image = self.original_image.rotate(self.rotation_angle)
self.display_image(self.rotated_image)

def save_rotated_image(self):
# Menyimpan gambar yang sudah dirotasi
if self.rotated_image:
save_path = filedialog.asksaveasfilename(defaultextension=".png", filetypes=[("PNG files", "*.png")])
self.rotated_image.save(save_path)
print("Rotated image saved successfully.")

def main():
root = tk.Tk()
app = ImageRotatorApp(root)
root.mainloop()

if __name__ == "__main__":
main()
```

Berikut ini adalah hasil dari program ini :



#### 4. KESIMPULAN

Penelitian ini mengungkapkan bahwa penggunaan metode Affine Transformation dalam rotasi gambar terbukti efektif dan efisien. Affine Transformation memungkinkan transformasi geometris yang meliputi translasi, skala, rotasi, dan shear, sehingga mampu memutar gambar tanpa mengubah bentuk objek. Implementasi menggunakan Python dan OpenCV menunjukkan bahwa metode ini dapat diterapkan dengan baik, menjaga kualitas dan detail gambar setelah rotasi serta menganalisis distorsi dan artefak yang mungkin muncul. Fleksibilitas Python sebagai bahasa pemrograman serbaguna dan kemampuannya dalam integrasi dengan sistem basis data memudahkan pengembangan aplikasi secara cepat dan reliabel. OpenCV, dengan berbagai algoritma dan fungsi pengolahan citra yang efisien, menjadi pilihan utama untuk implementasi teknik rotasi gambar ini. Proses implementasi yang mencakup instalasi OpenCV dan pengembangan aplikasi Image Rotation berhasil menunjukkan bahwa Affine Transformation dapat diterapkan untuk menghasilkan rotasi gambar yang presisi dan berkualitas. Aplikasi ini sangat relevan dalam bidang computer vision, pengolahan gambar medis, dan sistem keamanan visual, membantu pengguna dalam melakukan berbagai kegiatan yang berkaitan dengan gambar dengan lebih mudah dan cepat.

#### DAFTAR PUSTAKA

- Azhar, A. (2019). *Konsep Dasar Aplikasi*. Jakarta: Penerbit Sakti.
- Pahlevi, R., & Rosyani, P. (2021). Perancangan Sistem Informasi Penerimaan Peserta Didik Baru Berbasis Web Pada Mi Madinatunnjah. *JURIKOM (Jurnal Riset Komputer)*, 149–156.
- Ramzi, S. (2013). *Sistem Informasi Manajemen*. Yogyakarta: Penerbi Andi.
- Rosyani, P. (2023). Analisis Metode Forward Chaining pada Sistem Pakar: Systematic Literature Review. *Jurnal Manajemen, Ekonomi, Hukum, Kewirausahaan, Kesehatan, Pendidikan dan Informatika (MANEKIN)*, 144-151.
- Siradjuddin, H. K. (2017). *Perangkat Lunak Aplikasi: Buku Pengantar*. Jakarta: Rineka.
- Siswanto, B. F., & Rosyani, P. (2021). Perancangan Sistem Informasi Peggajian Karyawan Pada Tb Blitar Berbasis User Centered Design. *Journal of Information System Research (JOSH)*, 7-17.
- Susim, T., & Darujati, C. (2021). Pengolahan Citra untuk Pengenalan Wajah (Face Recognition) Menggunakan Opencv. *Jurnal Syntax Admiration*, 534-545.
- Unknown. (2024, June 22). *Geometric Transformations of Images*. Retrieved from OpenCV: [https://docs.opencv.org/4.x/da/d6e/tutorial\\_py\\_geometric\\_transformations.html](https://docs.opencv.org/4.x/da/d6e/tutorial_py_geometric_transformations.html)