

MAKALAH

## **SIGNATURE RECOGNITION USING PYTHON**

*Untuk Memenuhi Tugas Sebagian dari Mata Kuliah Pengolahan Citra Digital*

**Dosen Pengampuh:**

MUHAMMAD THAHIRUDDIN, S,Si, M.Mat



Oleh:

**Moh. Ainorrosi      IST2104043**

**M. Riyas Rasyid      IST2104038**

**Kholili      IST2104034**

**Khoirul Umam      IST2104032**

**Miftahul Arifin      IST2104042**

**FAKULTAS TEKNIK**

**PRODI TEKNOLOGI INFORMASI**

**INSTITUT SAINS DAN TEKNOLOGI ANNUQAYAH**

**GULUK-GULUK SUMENEP**

**JANUARI 2024**

## KATA PENGANTAR

Tanda tangan manusia telah menjadi lambang otentikasi yang dikenal luas dalam berbagai aspek kehidupan sehari-hari, mulai dari transaksi keuangan hingga proses hukum. Namun, seiring dengan kemajuan teknologi, kebutuhan akan sistem otomatis pengenalan tanda tangan menjadi semakin mendesak. "Signature Recognition using Python" merupakan bidang penelitian yang menggabungkan kecerdasan komputasional, pengolahan citra, dan keamanan untuk menciptakan solusi modern dalam menghadapi tantangan otentikasi manual.

Penerapan teknologi Python dalam pengenalan tanda tangan membuka pintu untuk inovasi baru dalam pendekatan ini. Python, sebagai bahasa pemrograman yang luwes dan populer, menyediakan kerangka kerja yang ideal untuk mengembangkan model pengenalan tanda tangan yang canggih. Dengan memanfaatkan perpustakaan dan alat bantu pengembangan seperti OpenCV, TensorFlow, dan Scikit-learn, penelitian dalam domain ini menawarkan solusi yang efektif, efisien, dan mudah diimplementasikan.

Makalah ini bertujuan untuk menjelajahi berbagai aspek pengenalan tanda tangan menggunakan Python, termasuk pengumpulan dan pra-pemrosesan data, pembangunan model, hingga evaluasi kinerja sistem secara keseluruhan. Kami akan mendalami teknik-teknik terkini dalam pengolahan citra dan pembelajaran mesin, serta menjelaskan bagaimana integrasi Python dapat memberikan keunggulan dalam pengembangan solusi otentikasi modern. Dalam pembahasan makalah ini, kita akan membahas tantangan umum yang dihadapi dalam pengenalan tanda tangan manual, dan bagaimana pendekatan berbasis Python dapat menjadi jawaban atas kompleksitas tersebut. Diharapkan, eksplorasi ini tidak hanya memberikan kontribusi terhadap penelitian ilmiah, tetapi juga membuka peluang untuk implementasi praktis dalam skenario kehidupan nyata.

Dengan demikian, "Signature Recognition using Python" tidak hanya merupakan sebuah konsep, melainkan sebuah langkah progresif menuju sistem otentikasi yang lebih cerdas, andal, dan sesuai dengan tuntutan zaman.

Penulis

## DAFTAR ISI

<b>HALAMAN SAMPUL.....</b>	<b>i</b>
<b>KATA PENGANTAR.....</b>	<b>Error! Bookmark not defined.</b>
<b>DAFTAR ISI.....</b>	<b>Error! Bookmark not defined.</b>
<b>BAB I.....</b>	<b>Error! Bookmark not defined.</b>
<b>PENDAHULUAN .....</b>	<b>Error! Bookmark not defined.</b>
A. Latar Belakang Masalah .....	<b>Error! Bookmark not defined.</b>
B. Rumusan Masalah.....	<b>Error! Bookmark not defined.</b>
C. Tujuan Pembahasan .....	<b>Error! Bookmark not defined.</b>
<b>BAB II .....</b>	<b>Error! Bookmark not defined.</b>
<b>PEMBAHASAN .....</b>	<b>Error! Bookmark not defined.</b>
A. Definisi Umum .....	<b>Error! Bookmark not defined.</b>
B. Optimalisasi Performa Sistem .....	<b>Error! Bookmark not defined.</b>
C. Out Put Signature Recognition using Python ....	<b>Error! Bookmark not defined.</b>
<b>BAB III.....</b>	<b>Error! Bookmark not defined.</b>
<b>PENUTUP.....</b>	<b>Error! Bookmark not defined.</b>
A. Kesimpulan.....	<b>Error! Bookmark not defined.</b>
B. Saran-saran .....	<b>Error! Bookmark not defined.</b>
<b>DAFTAR PUSTAKA.....</b>	<b>Error! Bookmark not defined.</b>
<b>BERIKUT ADALAH CODINGANNYA .....</b>	<b>Error! Bookmark not defined.</b>

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang Masalah**

Perkembangan teknologi artificial intelligence (AI) telah mengubah banyak aspek kehidupan, termasuk di industri musik. Salah satu aplikasi menarik dari AI adalah kemampuannya dalam menghasilkan cover song secara otomatis. Namun, dengan adanya karya musik yang dihasilkan oleh AI, muncul pula pertanyaan tentang hak cipta dan keabsahan legalitas karya tersebut. Dalam konteks cover song, hak cipta sangat penting untuk melindungi karya asli dan memberikan penghargaan kepada penciptanya.

Meskipun AI dapat menghasilkan cover song yang sangat mirip dengan aslinya, tetap saja ada kebutuhan untuk memastikan bahwa legalitas hak cipta tetap terjaga dan identifikasi pencipta asli tetap terlacak. Dalam kerangka inilah, latar belakang masalah ini menyoroti perlunya eksplorasi lebih lanjut tentang nuansa syukur dalam Islam, serta dampaknya terhadap pandangan dan dialog antaragama. Melalui pemahaman yang lebih dalam tentang nilai-nilai keislaman ini, diharapkan dapat memberikan wawasan yang lebih kaya dalam membangun keberagaman dan menjalin hubungan positif antara komunitas-komunitas berbeda keyakinan.

Tanda tangan merupakan salah satu biometrik yang paling banyak digunakan sebagai tanda atau ciri untuk otentikasi orang ataupun dokumen.<sup>1</sup> Identifikasi tanda tangan dapat dipelajari dan diimplementasikan pada banyak aplikasi meliputi perbankan, kartu kredit validasi, serta sistem keamanan yang secara umum menggunakan tanda tangan sebagai acuan untuk validasi suatu identitas ataupun dokumen.

Pada kenyataannya sering terjadi kesalahan dalam proses pengenalan, akurasi kecil atau kurang akurat dan dibutuhkan waktu yang lama yang disebabkan oleh sistem pengecekan yang dilakukan secara manual dalam penggunaan tanda tangan sebagai acuan untuk validasi data, sehingga diperlukan sistem untuk mempercepat

---

<sup>1</sup> Bharadi, dan Kekre., 2010

proses validasi data tersebut yang dapat mengenali tanda tangan dengan teknologi pengenalan pola citra digital dengan akurasi yang baik.

## **B. Rumusan Masalah**

1. Apa yang di maksud dengan *Signature Recognition using Python* ?
2. Bagaimana mengoptimalkan performa sistem agar dapat diimplementasikan secara efisien dalam aplikasi keamanan dan otentikasi menggunakan teknologi Python?
3. Apa manfaat dari *Signature Recognition using Python*?

## **C. Tujuan Pembahasan**

1. Memahami definisi umum mengenai *Signature Recognition using Python*.
2. Memahami cara mengoptimalkan performa sistem agar dapat diimplementasikan secara efisien dalam aplikasi keamanan dan otentikasi menggunakan teknologi Python.
3. Mengetahui manfaat dari *Signature Recognition using Python*.

## **BAB II**

### **PEMBAHASAN**

#### **A. Definisi Umum**

*Signature Recognition using Python* adalah suatu bidang penelitian yang bertujuan untuk mengembangkan sistem otomatis yang mampu mengenali dan memverifikasi tanda tangan manusia menggunakan bahasa pemrograman Python. Pendekatan ini melibatkan pemanfaatan teknologi pengolahan citra dan machine learning untuk meningkatkan akurasi serta efisiensi proses otentikasi tanda tangan manual.

Penelitian ini berfokus pada analisis dan implementasi berbagai metode dan teknik dalam pengenalan tanda tangan, dengan tujuan utama menciptakan sistem yang dapat diandalkan dalam mendeteksi ciri-ciri unik dari tanda tangan setiap individu. Dalam implementasinya, Python digunakan sebagai bahasa pemrograman utama, memanfaatkan berbagai pustaka dan framework yang mendukung pengolahan citra, pembelajaran mesin, dan pengembangan model.

Langkah-langkah penelitian melibatkan pengumpulan dataset tanda tangan, pra-pemrosesan data untuk normalisasi dan pembersihan citra, serta pengembangan model pengenalan tanda tangan. Proses evaluasi kinerja sistem dilakukan untuk mengukur sejauh mana keberhasilan model dalam mengenali dan memverifikasi tanda tangan dengan akurasi yang tinggi.

Penelitian ini memiliki relevansi yang signifikan dalam konteks keamanan dan otentikasi, di mana pengenalan tanda tangan yang akurat dapat digunakan dalam berbagai aplikasi seperti transaksi keuangan, dokumen legal, dan sistem keamanan. Dengan menggabungkan teknologi Python dan konsep-konsep terkini dalam pengolahan citra dan machine learning, penelitian ini diharapkan dapat memberikan solusi inovatif untuk meningkatkan keamanan dan efisiensi dalam proses otentikasi tanda tangan manual.

#### **B. Optimalisasi Performa Sistem**

Optimalisasi performa sistem merujuk pada serangkaian langkah dan tindakan yang diambil untuk meningkatkan efisiensi, kinerja, dan responsivitas sistem secara keseluruhan. Dalam konteks "Signature Recognition using Python," optimalisasi

performa sistem bertujuan untuk memastikan bahwa pengenalan tanda tangan dilakukan dengan cepat, akurat, dan tanpa menghabiskan sumber daya yang berlebihan

Mengoptimalkan performa sistem pengenalan tanda tangan menggunakan Python dapat dilakukan melalui beberapa strategi dan praktik terbaik. Berikut adalah beberapa langkah yang dapat diambil untuk meningkatkan efisiensi dan kinerja sistem dalam aplikasi keamanan dan otentikasi:

#### 1. Pemilihan Model Ringan:

Memilih model machine learning yang ringan dan sesuai dengan kebutuhan aplikasi. Model yang kompleks mungkin memerlukan sumber daya komputasi yang lebih besar, sedangkan model yang lebih sederhana dapat memberikan keseimbangan antara akurasi dan kinerja.

#### 2. Penggunaan Teknologi Terkini:

Memanfaatkan teknologi terkini dalam pengolahan citra dan machine learning yang dioptimalkan untuk Python, seperti TensorFlow atau PyTorch. Perbarui pustaka dan framework secara berkala untuk memanfaatkan peningkatan kinerja dan optimasi yang mungkin diterapkan oleh komunitas pengembang.

#### 3. Pengelolaan Memori dengan Baik:

Memastikan pengelolaan memori yang efisien. Menghindari alokasi memori yang tidak perlu dan membersihkan sumber daya setelah penggunaan dapat membantu mencegah kebocoran memori yang dapat mempengaruhi kinerja sistem.

#### 4. Penggunaan Teknik Pemrosesan Citra Cepat:

Mengoptimalkan proses pra-pemrosesan citra dengan menggunakan teknik pemrosesan citra cepat dan efisien. Hindari operasi yang kompleks dan memilih metode yang sesuai dengan kebutuhan aplikasi.

#### 5. Parallelisasi dan Distribusi Tugas:

Jika memungkinkan, pertimbangkan untuk menggunakan teknik parallelisasi atau distribusi tugas untuk memproses beberapa tanda tangan

secara bersamaan. Ini dapat membantu meningkatkan kecepatan dan respons sistem.

#### 6. Optimasi Hyperparameter:

Melakukan optimasi pada hyperparameter model machine learning. Eksperimen dengan parameter seperti ukuran batch, tingkat pembelajaran, dan jumlah epoch untuk mencari konfigurasi yang memberikan keseimbangan terbaik antara akurasi dan kinerja.

#### 7. Kompilasi Model:

Jika menggunakan framework seperti TensorFlow, pertimbangkan untuk mengompilasi model menggunakan teknik Just-In-Time (JIT) atau AOT (Ahead-Of-Time) compilation. Ini dapat meningkatkan kinerja model saat dijalankan.

#### 8. Monitoring dan Profiling:

Rutin memonitor dan melakukan profiling aplikasi untuk mengidentifikasi bagian-bagian yang memerlukan perhatian lebih. Alat-alat seperti cProfile dan memory profiler dapat membantu mengidentifikasi titik-titik bottleneck.

#### 9. Penggunaan Caching:

Menggunakan caching untuk menyimpan hasil operasi yang dapat di-cache, seperti hasil pra-pemrosesan atau hasil klasifikasi sebelumnya, dapat mengurangi waktu komputasi.

#### 10. Optimasi I/O:

Meminimalkan operasi I/O yang tidak perlu dan memastikan bahwa alur kerja sistem didesain untuk mengurangi jumlah panggilan ke penyimpanan eksternal.

#### 11. Penggunaan Pustaka Terkompilasi:

Menggunakan pustaka atau fungsi Python yang telah terkompilasi dapat meningkatkan kinerja dibandingkan dengan implementasi murni dalam Python.



Dengan menerapkan strategi-strategi ini tentu dapat meningkatkan efisiensi dan kinerja sistem pengenalan tanda tangan menggunakan Python dalam aplikasi keamanan dan otentikasi. Diperlukan pendekatan holistik dan pengujian untuk memastikan bahwa setiap langkah menghasilkan peningkatan yang diharapkan tanpa mengorbankan akurasi sistem.

### ***C. Out Put Signature Recognition using Python***

Penerapan Signature Recognition menggunakan Python dapat memberikan berbagai manfaat, terutama dalam konteks keamanan, otentikasi, dan efisiensi proses. Berikut adalah beberapa manfaat dari implementasi Signature Recognition menggunakan Python:

#### **1. Keamanan Dokumen dan Transaksi**

Meningkatkan keamanan dokumen dan transaksi dengan memastikan bahwa tanda tangan yang digunakan adalah sah dan berasal dari individu yang seharusnya.

#### **2. Pencegahan Pemalsuan**

Membantu mencegah pemalsuan tanda tangan, karena model yang dikembangkan dapat mendeteksi variasi penulisan dan ciri-ciri unik dari tanda tangan yang sulit dipalsukan.

#### **3. Otentikasi Pengguna**

Digunakan sebagai metode otentikasi pengguna dalam berbagai konteks, termasuk akses ke sistem, aplikasi perbankan, atau ruang keamanan yang memerlukan identifikasi individu.

#### **4. Efisiensi dan Automasi**

Meningkatkan efisiensi dan mengurangi ketergantungan pada proses manual dalam otentikasi tanda tangan, sehingga mengoptimalkan waktu dan sumber daya.

## 5. Integrasi dengan Sistem Modern

Menggunakan Python memungkinkan integrasi yang baik dengan sistem modern, termasuk teknologi pengolahan citra dan machine learning, yang dapat meningkatkan akurasi dan adaptabilitas sistem.

## 6. Penerapan pada Berbagai Bidang

Dapat diterapkan dalam berbagai bidang seperti keuangan, hukum, dan layanan publik, memberikan dampak positif pada otentikasi dan keamanan data di berbagai sektor.

## 7. Pengurangan Risiko Kecurangan

Dengan menggunakan teknologi pengenalan tanda tangan, risiko kecurangan dapat dikurangi, karena sistem dapat mendeteksi tanda tangan palsu atau tidak sah dengan lebih baik daripada metode manual.

## 8. Kemudahan dalam Verifikasi

Memberikan kemudahan dalam proses verifikasi tanda tangan, yang dapat bermanfaat dalam situasi-situasi yang memerlukan verifikasi identitas cepat dan akurat.

## 9. Peningkatan Efisiensi Prosedur Hukum

Dalam bidang hukum, implementasi Signature Recognition dapat meningkatkan efisiensi prosedur hukum yang melibatkan otentikasi dokumen dan kontrak.

## 10. Pengurangan Biaya dan Waktu

Dengan mengurangi ketergantungan pada proses manual dan meminimalkan risiko kesalahan manusia, implementasi ini dapat membantu mengurangi biaya dan waktu yang diperlukan untuk otentikasi tanda tangan.

## 11. Peningkatan Pengalaman Pengguna

Menyediakan pengalaman pengguna yang lebih baik dengan meminimalkan proses otentikasi yang rumit dan memastikan bahwa tanda tangan dapat diverifikasi dengan cepat dan andal.

## 12. Adaptabilitas dan Fleksibilitas

Memiliki kemampuan untuk beradaptasi dengan variasi dalam gaya penulisan tanda tangan dan dapat diterapkan pada skenario otentikasi yang berbeda dengan fleksibilitas.

Dengan manfaat-manfaat tersebut, implementasi Signature Recognition menggunakan Python dapat menjadi solusi yang efektif dan inovatif untuk meningkatkan keamanan dan efisiensi dalam berbagai konteks aplikasi.

## **BAB III**

### **PENUTUP**

#### **A. Kesimpulan**

Dalam eksplorasi mengenai "Signature Recognition using Python," dapat disimpulkan bahwa implementasi teknologi Python dalam pengenalan tanda tangan membuka jendela menuju masa depan otentikasi yang lebih efisien dan andal. Makalah ini menggali berbagai aspek, mulai dari pengumpulan data hingga evaluasi kinerja sistem, memberikan wawasan mendalam tentang tantangan dan solusi dalam membangun model pengenalan tanda tangan yang efektif.

Pentingnya keamanan dokumen dan transaksi memberikan landasan kuat bagi penelitian ini. Sistem yang mampu mengenali dan memverifikasi tanda tangan dengan tingkat akurasi yang tinggi memiliki dampak positif dalam berbagai sektor, mulai dari sektor keuangan hingga dunia hukum. Implementasi teknologi Python, dengan dukungan pustaka-pustaka kelas dunia, memberikan kemampuan adaptasi yang luar biasa terhadap perubahan kebutuhan dan dinamika yang ada.

Dengan demikian, implementasi tanda tangan menggunakan Python bukan hanya representasi teknologi modern, tetapi juga upaya untuk membawa otentikasi ke tingkat baru, menggabungkan kecerdasan komputasional dengan kemudahan pengembangan Python. Sebagai bagian dari inovasi keamanan informasi, penelitian ini menandai langkah penting dalam menghadirkan solusi otentikasi yang lebih cerdas dan dapat diandalkan di era digital saat ini.

#### **B. Saran-saran**

Untuk memperkaya makalah ini, disarankan untuk mendalami lebih lanjut beberapa studi kasus yang berhasil menerapkan system tersebut. Dengan merinci aplikasi konkret dari pendekatan ini dalam konteks penelitian tertentu, makalah dapat memberikan wawasan yang lebih kaya dan relevan.

## DAFTAR PUSTAKA

- Andri Z., Mohamad I., dan Wisnu U, (2016): Implementasi Algoritma Ant Colony Optimization Pada Aplikasi Lokasi Tempat Ibadah Terdekat Dikota Bandung, JOIN, L 3 . 5,
- Apniia A. A., Catur B. W., dan Kusworo Adi. (2017): Perancangan sistem Wdentifikasi biometrik jari tangan menggunakan Laplacian of Gaussian dan c@ktraksi kontur, Youngster Physics Journal, 6, 304 - 314.
- Ca-Tian Z., dan An-Xin Z. (2013): Using adaptive ant colony algorithm optimized BP neural network to identify the DGA fault, /EEE.
- Candra D., dan M. Muslikh. (2013): Perbandingan Akurasi Backpropagation Neural Nctwork dan ANFIS Untuk Memprediksi Cuaca, Journal of Scienufic Modeling & Computaton, 1, 2 - 4.
- Euss N., Supriyadi U., dan Tresna L.. (2018): Penerapan Algoritma Ant Colony Optmizatton Menentukan Nilai Optimal Dalam Memilih Objek Wisata Berbasis Android, S/MATRIS, 9, 2 - 4. f |

## **BERIKUT ADALAH CODINGANNYA**

```
import cv2

import numpy as np

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score


def preprocess_image(image_path):

    img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

    resized_img = cv2.resize(img, (100, 100)) # Ubah ukuran gambar menjadi
        100x100 piksel

    flattened_img = resized_img.flatten() # Flatten gambar menjadi array 1D

    return flattened_img


signature_samples = [preprocess_image('path/to/signature_image1.jpg'),

                     preprocess_image('path/to/signature_image2.jpg'),

                     ...]


non_signature_samples = [preprocess_image('path/to/non_signature_image1.jpg'),

                        preprocess_image('path/to/non_signature_image2.jpg'),

                        ...]


labels_signature = np.ones(len(signature_samples))

labels_non_signature = np.zeros(len(non_signature_samples))
```

```
X = np.concatenate((signature_samples, non_signature_samples), axis=0)
```

```
y = np.concatenate((labels_signature, labels_non_signature), axis=0)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,  
                                                    random_state=42)
```

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
print(f'Akurasi Model: {accuracy}')
```

```
new_signature = preprocess_image('path/to/new_signature_image.jpg')
```

```
prediction = model.predict([new_signature])
```

```
if prediction[0] == 1:
```

```
    print("Tanda Tangan Dikenali")
```

```
else:
```

```
    print('Bukan Tanda Tangan')
```