

## **IDENTIFIKASI KEMIRIPAN FOTO ASLI DAN SKETSA MENGGUNAKAN MODEL GENERATIVE ADVERSARIAL NETWORKS (GANs)**

**Andre Satriawan<sup>1</sup>, Bahtiar Imran\*<sup>2</sup>, Surni Erniwati<sup>3</sup>**

<sup>1</sup>Teknik Informatika, Fakultas Teknik Informatika dan Komunikasi, Universitas Teknologi Mataram, Indonesia

<sup>2</sup>Rekayasa Sistem Komputer, Fakultas Teknik Informatika dan Komunikasi, Universitas Teknologi Mataram, Indonesia

<sup>3</sup>Manajemen Informatika, Fakultas Vokasi, Universitas Teknologi Mataram, Indonesia

Email: <sup>1</sup>andrelainan@gmail.ac.id, <sup>2</sup>bahtiarimranlombok@gmail.ac.id, <sup>3</sup>mentari1990@gmail.ac.id

(Naskah masuk : 20 Juni 2023, Revisi : 11 September 2023, Diterbitkan : 20 September 2023)

### **Abstrak**

Perkembangan seni semakin bertumbuh khususnya dalam bidang seni lukis, pertumbuhan tersebut terlihat dari banyaknya pemula yang mulai belajar melukis secara otodidak diawali dengan belajar membuat sketsa menggunakan metode yang beragam, tetapi masalah umum yang sering dihadapi oleh pemula dalam seni Lukis adalah seringkali sketsa dan foto asli terlihat serupa tetapi tidak tahu seberapa mirip sketsa yang telah dibuat. Penelitian ini bertujuan untuk mengidentifikasi persentase kemiripan foto asli dan sketsa menggunakan metode diskriminatif dari model Generative Adversarial Networks (GANs) memantkan library atau modul ssim. Diskriminator merupakan CNN yang menerima input gambar berukuran sama atau memiliki dimensi yang sama dan menghasilkan angka yang menyatakan apakah input merupakan gambar yang sama atau memiliki kemiripan. Untuk mendapatkan persentase kemiripan yang tepat antara dua gambar memanfaatkan Struktural Similarity Index (SSIM) yang telah terlatih pada library scikit-image.

**Kata kunci:** gambar, generative adversarial networks (gans), sketsa dan foto asli, struktural similarity index (ssim).

## **IDENTIFYING SIMILARITY TO ORIGINAL PHOTOS AND SKETCHES USING A GENERATIVE ADVERSARIAL NETWORKS (GANs) MODEL**

### **Abstract**

*Abstrak berbahasa Inggris diletakkan pada bagian ini. Gunakan font Times New Roman 10pt, italic. The development of art is growing, especially in the field of painting, this growth can be seen from the number of beginners who are starting to learn to paint by themselves, starting with learning to make sketches using various methods, but a common problem often faced by beginners in painting is that the original sketches and photos are often visible. similar but don't know how similar the sketch has been made. This research aims to identify the percentage of similarities between original photos and sketches using a discriminative method from the Generative Adversarial Networks (GANs) model using a library or ssim module. The discriminator is a CNN that receives input images of the same size or have the same dimensions and produces a number that states whether the input is the same image or has similarities. To get the correct percentage of similarity between two images, use the Structural Similarity Index (SSIM) which has been trained on the scikit-image library.*

**Keywords:** images, generative adversarial networks (gans), original sketches and photos, structural similarity index (ssim)

---

### **1. PENDAHULUAN**

Perkembangan ilmu computer vision dalam beberapa tahun terakhir telah membantu manusia dalam menyelesaikan tugasnya, seperti contohnya dalam menggunakan pengenalan wajah sebagai sistem keamanan maupun pencarian. *Face Recognition* pada umumnya merupakan sebuah proses untuk mengidentifikasi maupun memverifikasai identitas seseorang dengan menggunakan wajahnya. Penggunaan face recognition telah diterapkan oleh pihak otoritas hukum dalam mencari tersangka tindak pidana [1]. Perkembangan teknologi dalam bidang klasifikasi citra menggunakan *Generative Adversarial Networks (GANs)* salah satu algoritma *deep learning* dan salah satu cabang dari ilmu *artificial intelligence* telah mengalami peningkatan yang sangat pesat dan memberikan dampak signifikan di berbagai bidang [2].

Penggunaan GANs dalam pembuatan data citra baru, khususnya dalam menciptakan gambar sketsa yang serupa dengan gambar acuan, telah menjadi hal umum. Proses ini melibatkan kombinasi sampel seperti titik, garis, bidang, dan warna untuk menciptakan sketsa atau gambar imitasi dari objek tertentu, seperti objek fisik atau manusia. Proses ini mirip dengan pembuatan sketsa manual oleh seniman di bidang lukisan [3].

Namun, pembuatan sketsa manual oleh pemula dalam bidang seni yang belajar secara otodidak sering menghadapi tantangan dalam mencapai tingkat kemiripan yang memadai antara foto asli dan sketsa yang dibuat. Untuk mengatasi masalah tersebut, penelitian ini menggunakan metode diskriminatif dari model GANs [4] dan memanfaatkan fungsi *Structural Similarity Index (SSIM)* yang dapat mengukur sejauh mana struktur dan komponen dari dua gambar mirip satu sama lain. Dengan menggunakan model GANs yang menggabungkan elemen SSIM [5], penelitian ini bertujuan untuk mengukur kemiripan foto asli yang menjadi acuan dan sketsa berdasarkan proses pelatihan data, dengan tujuan mencapai akurasi seberapa miripnya sketsa yang dihasilkan.

## 2. METODE PENELITIAN

Metode penelitian yang dilakukan pada penelitian ini adalah penentuan tingkat kedekatan atau persentase kemiripan antara foto asli dan sketsa penulis melakukan menggunakan model *Generative Adversarial Networks (GANs)* yang bersifat diskriminatif. Dalam metode ini penulis memanfaatkan fungsi *Structural Similarity Index (SSIM)* sebagai alat ukur untuk menghitung persentase kemiripan antara foto asli dan sketsa.

### 2.1 Sumber Data

Gambar dibawah ini adalah sketsa kasar dan foto asli yang akan diidentifikasi. dataset sketsa kasar sendiri di peroleh dari hasil pembuatan sketsa secara manual dan sketsa yang dibuat menggunakan *GANs* berdasarkan foto asli yang akan di jadikan sebagai dataset untuk melakukan proses identifikasi perbandingan kemiripan foto asli dan sketsa kasar [6], [7]. Proses identifikasi ini peneliti menggunakan foto asli dari beberapa sumber terdiri dari *google image, intagram, pinterest, Kaggle* dan yang terakhir bersumber dari peneliti sendiri. Dataset ini diperlukan untuk menguji model apakah benar-benar bisa menghasilkan persentase yang akurat. Gambar diatas merupakan salah satu contoh sketsa kasar dan foto asli yang menggunakan *format file .jpg* dan *.png*.



Gambar 1. Sketsa dan foto asli

### 2.2 Arsitektur Umum Perancangan Antarmuka Model

Arsitektur Umum Perancangan Antarmuka Sistem dan proses pengujian Identifikasi ini semuanya di kerjakan pada aplikasi berbasis *online* [8] yang di sediakan oleh *google* yang biasa disebut dengan nama *google colab* atau *google colabatory*. Antarmuka sistem biasa disebut dengan *feature extraction and reverse image search* yang terdiri terdiri dari beberapa bagian yaitu :

#### a. *Import library*

Gambar dibawah merupakan baris kode berupa modul *import* pada *python* yang berfungsi untuk *multi file* yang dimaksud dengan *multi file* disini dapat memanggil file lain di dalam satu model yang berbeda atau dengan penjelasan sederhananya untuk menginstal beberapa *library* yang akan mendukung proses identifikasi [6].

```
1] from skimage.metrics import structural_similarity as ssim
from skimage.metrics import structural_similarity
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt
import cv2
import numpy as np
#import face_recognition
import imutils #imutils includes opencv functions
from imutils import paths
import pickle
import time
import os
```

Gambar 2. Import library

### b. Modeling Diskriminator

Kode dibawah ini merupakan sebuah definisi model neural network yang menggunakan TensorFlow Sequential API. Model ini terdiri dari beberapa layer yang diatur secara berurutan. Pertama, terdapat layer Conv2D yang bertujuan untuk mengekstraksi fitur dari gambar dengan menggunakan filter berukuran 3x3 dan fungsi aktivasi ReLU. Kemudian, terdapat layer MaxPooling2D yang berfungsi untuk mengurangi dimensi spasial dari fitur yang telah diekstraksi sebelumnya. Selanjutnya, ada layer Conv2D dan MaxPooling2D yang diulang sekali lagi untuk melakukan ekstraksi dan pengurangan dimensi spasial lebih lanjut. Setelah itu, terdapat layer Flatten yang digunakan untuk meratakan output dari layer sebelumnya menjadi satu dimensi. Akhirnya, terdapat dua layer Dense, di mana layer pertama memiliki 1064 neuron dengan fungsi aktivasi ReLU, dan layer terakhir memiliki 2 neuron dengan fungsi aktivasi sigmoid yang menghasilkan prediksi klasifikasi biner. Model ini digunakan untuk memproses gambar dengan ukuran 150x150 piksel dan menghasilkan prediksi terkait klasifikasi yang dilakukan. [6]

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), padding='same', activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(128, (3, 3), padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(1064, activation='relu'),
    tf.keras.layers.Dense(2, activation='sigmoid')
])
```

Gambar 3. Modeling Diskriminator

### c. Data Aquition

Kode untuk memanggil dataset yang sebelumnya sudah disiapkan dan disimpan pada google drive. Hal ini perlu dilakukan karena jika dataset lansung di impor dari penyimpanan lokal kedalam lembar kerja pada google colab kemungkinan besar akan terhapus secara otomatis jika ada masalah dengan sambungan ineternet. Memanggil dataset dari google drive juga akan mengalami hal serupa yang membedakan file akan terhapus jika google colab ditutup dan tidak akan terhapus apabila sambungan internet bermasalah. Impor dataset dari google drive juga tidak membutuhkan waktu yang lama. Pada saat melakukan impor file dari lokal akan membutuhkan waktu yang cukup lama jika file berukuran besar.[9]

```
[ ] from google.colab import drive
drive.mount('/content/drive')
```

Gambar 4. import dataset

### d. Load Image dan Diskriminator

kode untuk memanggil atau memuat gambar dan model diskriminator dari *google drive* yang akan diidentifikasi. setelah itu gambar yang sudah di *input* akan dilakukan perubahan dimensi agar bisa terbaca oleh *ssim*

```
# Muat model diskriminator yang telah dilatih
diskriminator = tf.keras.models.load_model('/content/discriminator.h5')

first = cv2.imread('/content/drive/MyDrive/sketsa/Test/IU real.png.jpg')
second = cv2.imread('/content/drive/MyDrive/sketsa/Test/IU sket.jpg')

width = 800
height = 800
dim = (width,height)
resized_FIRST = cv2.resize(first, dim, interpolation = cv2.INTER_AREA)

width = 800
height = 800
dim = (width,height)
resized_SECOND = cv2.resize(second, dim, interpolation = cv2.INTER_AREA)
```

Gambar 5. load image dan diskriminator

#### e. Forwarding image

Perintah atau sebuah fungsi untuk merubah gambar berwarna menjadi abu-abu sebelum di eksekusi oleh `ssim`. `cv2.imread` berfungsi untuk menampilkan gambar yang diidentifikasi, dan perintah `print("Skor kemiripan: {:.3f}%" .format(score * 100))` berfungsi untuk menampilkan persentase keemiripn yang di dapatkan dari dua gambar yang diidentifikasi.[3].

```
first_gray = cv2.cvtColor(resized_FIRST, cv2.COLOR_BGR2GRAY)
second_gray = cv2.cvtColor(resized_SECOND, cv2.COLOR_BGR2GRAY)

score, diff = structural_similarity(first_gray, second_gray, full=True)
cv2.imshow(resized_FIRST)
cv2.imshow(resized_SECOND)
print("Skor kemiripan : {:.3f}%" .format(score * 100))
```

Gambar 6. Function menyamakan warna dan function `ssim`.

#### f. Output

Gambar dibawah ini untuk menampilkan *output* kemiripan pada gambar yang telah diidentifikasi, dengan mencari kontur yang memiliki kemiripan, bagian dari gambar yang memiliki kemiripan yang paling jelas akan ditandai dengan abu-abu pada gambar

```
from google.colab.patches import cv2_imshow
for c in contours:
    area = cv2.contourArea(c)
    if area > 100:
        x,y,w,h = cv2.boundingRect(c)
        cv2.rectangle(resized_FIRST, (x, y), (x + w, y + h), (36,255,12), 2)
        cv2.rectangle(resized_SECOND, (x, y), (x + w, y + h), (36,255,12), 2)

cv2_imshow(resized_FIRST)
cv2_imshow(resized_SECOND)
cv2_imshow(diff)
```

Gambar 7. Output

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Data testing dan Data training

Hasil dari epoch model discriminator gans, dalam penelitian ini menggunakan epoch 15. hasil akurasi pada epoch 1 mendapatkan akurasi sebesar 80,80% dan mendapatkan loss sebesar 39,60%, pada epoch 10 mendapatkan akurasi sebesar 98,66% dan mwndapatkan loss yang cukup besar yaitu sebesar 84,39%, hasil kurasi terbaik didapatkan pada epoch ke 15 sebesar 99,55% dan mendapatkan nilai loss yang sangat kecil, yaitu sebesar 02,70%.

Tabel 1. Hasil *Epoch Training* Diskriminator

Epoch	Loss	Accuracy
1	39,60%	80,80%
10	84,39%	98,66%
15	02,70%	99,55%

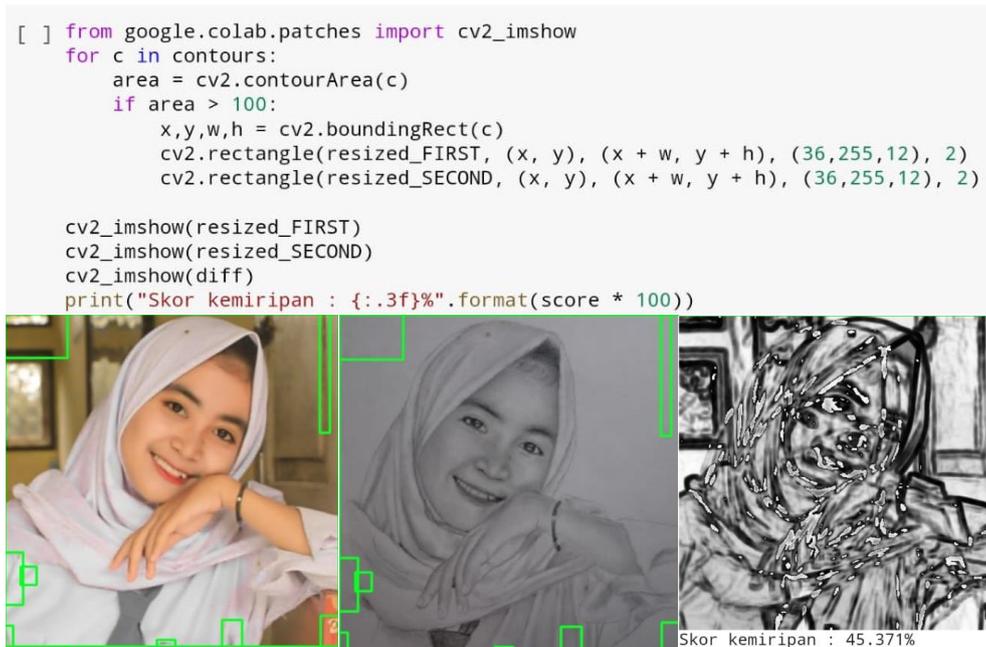
Gambar 8 menggambarkan cara kerja *ssim* yang telah terlatih untuk mengukur akurasi dan kualitas gambar. Skor *mse* 0,00 menunjukkan bahwa kualitas gambar memiliki kualitas yang sangat baik tanpa adanya pengurangan jumlah piksel. Namun, jika *mse* menunjukkan skor 3894,84, hal ini menandakan adanya jarak atau penurunan kualitas gambar dan. Skor *ssim* 0,65 mewakili persentase kemiripan foto asli dan sketsa, yaitu sebesar 65%. Pada penelitian ini, pengujian menggunakan *mse* tidak diterapkan, melainkan fokus pada penggunaan *ssim* untuk mengidentifikasi persentase kemiripan pada foto asli dan sketsa [9], [10].



Gambar 8. Testing accuracy *ssim*

### 3.2. Hasil penelitian

Gambar 9 adalah hasil identifikasi dari dua set data, berupa sketsa dan foto asli yang menjadi acuan. Persentase kemiripan sebesar 45.371% diperoleh dengan identifikasi menggunakan metode diskriminatif model *generative adversarial networks (GANs)* memanfaatkan fungsi *Struktural Similarity Index Measure (SSIM)*, model tersebut membaca kedua gambar yang di *input*, dan saling membandingkannya secara komparatif.



### Gambar 9. Hasil identifikasi

Persentase 45.371%. ditunjukkan oleh bagian yang memiliki struktur pembentuk yang mirip, ditandai dengan warna abu-abu pada gambar identifikasi diatas. bagian yang tidak memiliki struktur yang sama seperti ditunjukkan pada bagian kotak hijau diatas tidak akan terbaca oleh model, contohnya *background* tidak memiliki struktur yang serupa, yang terbaca hanya sruktur yang memiliki kesamaan. Bagian dengan tingkat keabuan atau bagian yang memiliki *blur* paling tinggi menunjukkan kemiripan yang lebih signifikan. Jika salah satu dari dua gambar yang di *input* memiliki kualitas resolusi yang rendah model akan menunjukkan persentase kemiripan 0.00%. karena model tidak dapat mengidentifikasi kemiripan antara kedua gambar tersebut gambar yang dapat terbaca hanya gambar yang memiliki kualitas resolusi yang cukup tinggi.

#### 4. KESIMPULAN

Penelitian ini memberikan kesimpulan bahwa model GAN tidak hanya digunakan untuk menciptakan gambar baru, tetapi juga dapat mengidentifikasi kemiripan antara gambar-gambar tersebut. Fungsi SSIM dalam diskriminator GAN digunakan untuk mengukur tingkat kemiripan antara foto asli dan sketsa. Dengan adanya penelitian ini, kita dapat memahami bahwa model GAN dengan menggunakan fungsi SSIM dapat memberikan informasi tentang sejauh mana kedua gambar tersebut mirip satu sama lain. Hal ini membantu dalam mengukur tingkat kemiripan antara foto asli dan sketsa yang dihasilkan.

#### DAFTAR PUSTAKA

- [1] M. A. Rasyid, R. F. Rachmadi, and D. P. Wulandari, "Pembangkitan Citra Wajah dari Sketch Wajah menggunakan CycleGAN," *J. Tek. ITS*, vol. 11, no. 3, 2022, doi: 10.12962/j23373539.v11i3.94440.
- [2] J. Arifianto and I. Muhimmah, "Aplikasi Web Pendeteksi Jerawat Pada Wajah Menggunakan Algoritma Deep Learning dengan TensorFlow," *J. Autom.*, pp. 21–29, 2021.
- [3] R. Reinaldo Kalendesang and D. Haryadi Setiabudi, "Pewarnaan Otomatis Sketsa Gambar Menggunakan Metode Conditional GAN Untuk Mempercepat Proses Pewarnaan," *J. Infra*, vol. 10, no. 2, 2022.
- [4] M. Kurniasih and S. Akbar, "Identifikasi Kemiripan Wajah Untuk Kehadiran Karyawan Menggunakan Algoritma Eigenface," *Respati*, vol. 13, no. 2, pp. 78–84, 2018, doi: 10.35842/jtir.v13i2.237.
- [5] B. Yanto, L. Fimawahib, A. Supriyanto, B. H. Hayadi, and R. R. Pratama, "Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan Metode Deep Learning Convolutional Neural Network," *INOVTEK Polbeng - Seri Inform.*, vol. 6, no. 2, p. 259, 2021, doi: 10.35314/isi.v6i2.2104.
- [6] M. Ricky and M. E. Al Rivan, "Implementasi Deep Convolutional Generative Adversarial Network untuk Pewarnaan Citra Grayscale," *J. Tek. Inform. dan Sist. Inf.*, vol. 8, no. 3, pp. 556–566, 2022, doi: 10.28932/jutisi.v8i3.5218.
- [7] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," *Adv. Neural Inf. Process. Syst.*, vol. 2017-December, no. Nips, pp. 6627–6638, 2017, doi: 10.18034/ajase.v8i1.9.
- [8] V. Nagarajan and J. Z. Kolter, "Gradient descent GAN optimization is locally stable," *Adv. Neural Inf. Process. Syst.*, vol. 2017-December, no. Nips, pp. 5586–5596, 2017.
- [9] R. Wang, A. Cully, H. J. Chang, and Y. Demiris, "MAGAN: Margin Adaptation for Generative Adversarial Networks," *arXiv e-prints*, no. April, 2017, [Online]. Available: <http://arxiv.org/abs/1704.03817>
- [10] M. R. Efrian and U. Latifa, "Image Recognition Berbasis Convolutional Neural Network (Cnn) Untuk Mendeteksi Penyakit Kulit Pada Manusia," *Power Elektron. J. Orang Elektro*, vol. 11, no. 2, p. 276, 2022, doi: 10.30591/polektron.v12i1.3874.