

DETEKSI JERAWAT DI WAJAH MENGGUNAKAN SEGMENTASI GAMBAR

ACNE DETECTION ON THE FACE USING IMAGE SEGMENTATION

¹Harapan Napitupulu, ²Dhani Ariatmanto

¹Universitas Amikom Yogyakarta , Jl.Ring Road Utara, Ngiringin, Condongcatu, Depok,
Sleman 55281 Indonesia

²Prodi Intelligent Animation Fakultas Teknik Informatika Universitas Amikom Yogyakarta, Jl.
Ring Road Utara, Ngiringin, Condongcatu, Depok, Sleman, 55281 INDONESIA

¹harapan_napitupulu@students.amikom.ac.id, ²dhaniari@amikom.ac.id,

Received:
13 Agustus 2025

Revised:
21 Oktober 2025

Accepted:
24 November 2025

Published:
7 Februari 2026

ABSTRAK

Jerawat merupakan suatu penyakit yang biasa terjadi di wajah bagi anak remaja atau dewasa atau juga disebut masa puberitas dari usia Sembilan tahun sampai delapan belas tahun keatas. Penyakit kulit ini sangat sulit disembuhkan walaupun tidak berdampak pada kesehatan fisik manusia, tapi sangat mempengaruhi mental psikologi jiwa manusia yang ingin tampil pede, bersih, rapih, cantik, dan tampan. Sudah banyak penelitian yang digunakan dalam mendeteksi jerawat dengan berbagai metode yang masih rumit dalam mendeteksi jerawat. Dalam pengobatan wajah berjerawat diperlukan pengidentifikasi jerawat yang cepat mudah dan sederhana dengan teknologi, agar hasil pengobatan dapat dilakukan secara maksimal dan biaya yang murah. Tujuan dari penelitian ini adalah untuk mendeteksi jerawat dengan menggunakan segmentasi gambar yang lebih mudah dan simple untuk dipahami dengan mengkompres gambar menggunakan algoritma K-means lalu disegmentasi. Segmentasi gambar adalah pembagian gambar digital menjadi kelompok piksel diskrit untuk mendeteksi objek dan klasifikasi objek. Hasil dari penelitian ini menunjukkan bahwa menggunakan metode segmentasi gambar mampu mendeteksi jerawat dengan identifikasi berbentuk titik dan lingkaran di wajah dengan mencocokkan gambar asli dengan gambar yang di segmentasikan. Maka dari itu dengan menggunakan metode ini dapat digunakan dan membantu dalam pengobatan wajah berjerawat menggunakan segmentasi gambar.

Kata Kunci : Segmentasi Gambar, Deteksi, Jerawat, K-means

ABSTRACT

Acne is a common facial disease for teenagers or adults, also known as puberty, from the age of nine to eighteen years and above. This skin disease is very difficult to cure, although it does not affect human physical health, but it greatly affects the mental psychology of the human soul who wants to appear confident, clean, neat, beautiful, and handsome. Many studies have been used in detecting acne with various methods that are still complicated in detecting acne. In the treatment of acne-prone faces, a fast, easy, and simple acne identifier is needed with technology, so that treatment results can be carried out optimally and at a low cost. The purpose of this study is to detect acne using image segmentation that is easier and simpler to understand by compressing images using the K-means algorithm and then segmenting them. Image segmentation is the division of digital images into discrete pixel groups to detect objects and object classification. The results of this study show that using the image segmentation method is able to detect acne with identification in the form of dots and circles on the face by matching the original image with the segmented image. Therefore, using this method can be used and helps in the treatment of acne-prone faces using image segmentation.

Keywords : Image Segmentation, Detection, Acne, K-means.

PENDAHULUAN

Jerawat adalah masalah kulit yang dialami oleh setiap manusia pada masa puberitas umumnya pada anak remaja dari usia delapan sampai delapan belas tahun maupun Wanita ataupun laki laki dan tidak Cuma anak remaja saja, orang dewasa maupun orang tua juga mengalaminya. Banyak penelitian yang telah dilakukan untuk mendeteksi jerawat dengan berbagai metode, contohnya metode yang digunakan algoritma CNN, YOLO v5(Pinasty, Bagus, and Hakim 2025), Viola Jones dan Segmentasi citra. Tujuan dari penelitian ini untuk mendeteksi jerawat dengan menggunakan metode segmentasi gambar. Yang dimanah acuan penelitian ini diambil dari artikel A Method for Animating Children's Drawings of The Human Figure yang melakukan penelitian bagaimanah menggerakkan gambar anak-anak menjadi animasi yang menggunakan tiga tahap metode yaitu metode Detection, Segmentation, dan Pose Estimation on Non-photorealistic image. Peneliti mengambil salah satu metode yang akan di gunakan yaitu menggunakan segmentation gambar untuk mendeteksi jerawat(Smith et al. 2023).

Penelitian ini sangat sederhana tapi sangat penting untuk diteliti dengan metode segmentasi gambar. Dengan menggunakan metode ini peneliti bertujuan untuk melihat perbedaan melalui citra gambar dari gambar asli di estrak menjadi gambar abu abu untuk mendeteksi jerawat. Penelitian jerawat pernah dilakukan dengan metode segmentasi citra yaitu mendeteksi jerawat dengan pemetaan gambar RGB ke spektrum sebagai cara untuk Memvisualisasikan perubahan warna. Ini adalah satu satunya penelitian pemetaan RGB ke spektrum tampak dengan kasifikasi warna merah, ungu, dan coklat. Walaupun penelitian ini menggunakan metode segmentasi tapi penelitian ini menggunakan pencocokan gambar asli yang diubah menjadi Abu-abu dengan beberapa tahap kompres gambar menggunakan algoritma K-means.

Salah satu tantangan penelitian ini adalah mengkompres gambar dan di ubah menjadi abu-abu untuk mendeteksi jerawat dengan dermatologi seperti bintik, kerutan dan jerawat. Mengkompres gambar menggunakan algoritma python 3 dengan metode K-Means membutuhkan computer dengan CPU yang lebih tinggi. Mengkompres gambar menggunakan python dengan lebih teliti. Pencocokan gambar dengan yang asli dengan gambar yang dirubah menjadi abu-abu. Mendeteksi jerawat dengan memperbandingkan gambar asli dengan gambar kompres abu-abu.

METODE

Dataset Gambar

Data gambar berjerawat peneliti menggunakan data sekunder yang di unduh atau download di google image dengan kata kunci penelusuran wajah berjerawat. Berikut adalah contoh gambar sempel yang akan digunakan.



Gambar 1. Dataset sekunder wajah berjerawat

Mengkompres Gambar

Setelah mengumpulkan data gambar sebelum mengkompres peneliti akan menjelaskan perelatan yang digunakan. Dalam peneliti ini, penenliti menggunakan laptop lenovo Legion 5 dengan intel core i7 Gen 12 RTX 3050 Ti dengan RAM 16 GB dalam membantu pemograman menggunakan Python di Google Colab. Dalam mengkompres gambar peneliti menggunakan metode K-Means algoritma yang merupakan salah satu Mecine Learning. Mecine learning adalah pembelajaran mesin ilmu computer tanpa di program secara eksplisit(HASYDNA and DINATA 2020). Salah satu pembelajaran algoritma yang populer digunakan K-means clustering untuk mempartisi pengamatan kedalam klaster yang dimanah setiap observasi memiliki nilai rata rata centroid. Dalam penelitian ini peneliti menggunakan metode ini untuk mengekstrak gambar, dari gambar asli ke gambar yang lebih kontras. Pertama sebelum mengkompres gambar cara pertama yang harus dilakaukan membuka python di google colab karena peneliti menggunakan google colab dalam mengoprasikan python, setelah google colab dibuka langkah selanjutnya mengupload data file gambar berjerawat di google colab. Setelah data terkumpul maka jalankan pemograman seperti contoh gambar dibawah ini.

```
▶ from skimage import io
import numpy as np
import numpy.matlib

image = io.imread('/content/jerawt6.webp')
io.imshow(image)
io.show()
```

Gambar 2. Pemrograman pertama

Dari gambar di teks berwarna merah merupakan data gambar Copy Path yang diupload di google colab, setelah gambar pemrograman diatas dijalankan maka akan muncul gambar sebagai berikut.



Gambar 3. Hasil Pemrograman Pertama

Setelah gambar diatas agar file gambar di google colab dapat terbaca maka langkah selanjutnya menjalankan pemrograman seperti dibawah ini.

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

↳ Mounted at /content/drive
```

Gambar 4. Agar file di google colab dapat terbaca

Dalam mengkompres gambar peneliti menggunakan pixel 255 untuk menentukan nilai RGB nya.

```
▶ rows = image.shape[0]
cols = image.shape[1]
image = image/255
X = image.reshape([image.shape[0]*image.shape[1],3])
```

Gambar 5. Nilai pixel yang digunakan

Setelah nilai pixel sudah ditentukan maka langkah selanjutnya menentukan jumlah kelompok atau Clucter dan jumlah iterasi maksimum yaitu dengan K=16 dan iterasinya 50 seperti pemrograman dibawah ini.

```
▶ K = 16 #number of clusters
max_iters = 50 #number of times the k-mean should run
```

Gambar 6. Menentukan jumlah cluster dan iterasinya

Lalu jalankan pemrograman di atas dan langkah selanjut menghitung titik pusat awal centroid seara acak menggunakan algoritma K-means Clustering seperti dibawah ini.

```
▶ import random
def init_centroids(X,K):
    c = random.sample(list(X),K)
    return c
```

Gambar 7. Menentukan titik awal centroid

Jika titik pusat awal centroid sudah ditentukan maka lankah selajutnya adalah menentukan centroid terdekat dalam algoritma K-means.

```
def closest_centroids(X,c):  
    K = np.size(c,0)  
    idx = np.zeros((np.size(X,0),1))  
    arr = np.empty((np.size(X,0),1))  
    for i in range(0,K):  
        y = c[i]  
        temp = np.ones((np.size(X,0),1))*y  
        b = np.power(np.subtract(X,temp),2)  
        a = np.sum(b,axis = 1)  
        a = np.asarray(a)  
        a.resize((np.size(X,0),1))  
        #print(np.shape(a))  
        arr = np.append(arr, a, axis=1)  
    arr = np.delete(arr,0,axis=1)  
    idx = np.argmin(arr, axis=1)  
    return idx
```

Gambar 8. Menentukan Centroid terdekat

Langkah selanjutnya menghitung ulang posisi centroid di K-means Clustering seperti algoritma dibawah ini.

```
def compute_centroids(X,idx,K):  
    n = np.size(X,1)  
    centroids = np.zeros((K,n))  
    for i in range(0,K):  
        ci = idx==i  
        ci = ci.astype(int)  
        total_number = sum(ci)  
        ci.resize((np.size(X,0),1))  
        total_matrix = np.matlib.repmat(ci,1,n)  
        ci = np.transpose(ci)  
        total = np.multiply(X,total_matrix)  
        centroids[i] = (1/total_number)*np.sum(total,axis=0)  
    return centroids
```

Gambar 9. Menghitung ulang posisi centroid

Setelah menghitung ulang posisi centroid dilakukan maka langkah selanjutnya mengimplementasikan algoritma K-means untuk pengelompokan data dan menghasilkan koordinat centroid baru seperti pemrograman dibawah ini.

```
def run_kMean(X,initial_centroids,max_iters):  
    m = np.size(X,0)  
    n = np.size(X,1)  
    K = np.size(initial_centroids,0)  
    centroids = initial_centroids  
    previous_centroids = centroids  
    idx = np.zeros((m,1))  
    for i in range(1,max_iters):  
        idx = closest_centroids(X,centroids)  
        centroids = compute_centroids(X,idx,K)  
    return centroids,idx
```

Gambar 10. Mengimplementasikan K-means

Maka langkah selanjutnya menentukan nilai centroidnya dengan menjalankan pemrograman dibawah ini

```
initial_centroids = init_centroids(X,K)  
centroids,idx = run_kMean(X,initial_centroids,max_iters)  
#idx.resize((np.size(X,0),1))  
print(np.shape(centroids))  
print(np.shape(idx))
```

(16, 3)
(667000,)

Gambar 11. Nilai titik Centroid

Dari gambar diatas dapat disimpulkan bahwa ada 16 centroid dengan tiga fitur dengan jumlah idx nya 667.000 data. Langkah selanjutnya menggunakan pemrograman dibawah ini untuk mengkompres gambar.

```
idx = closest_centroids(X,centroids)
X_recovered = centroids[idx]
print(np.shape(X_recovered))
X_recovered = np.reshape(X_recovered, (rows, cols, 3))
print(np.shape(X_recovered))
```

Gambar 12. Mengkompres gambar Menggunakan K-means

Dari algoritma diatas digunakan untuk mengkompres gambar untuk menentukan tinggi, lebar dan warna RGBnya. Langkah selanjutnya setelah gambar di kompres maka menyimpan array gambar seperti pemograman dibawah ini.

```
[11] from keras.preprocessing.image import save_img
save_img('jerawt6.webp', X_recovered)
```

Gambar 13. Menyimpan gambar hasil kompres

Setelah menyimpan gambar telah dilakukan maka langkah selanjutnya memunculkan hasil gambar yang telah di kompres.

```
image_compressed = io.imread('jerawt6.webp')
io.imshow(image_compressed)
io.show()
```

Gambar 14. Memunculkan gambar hasil kompres

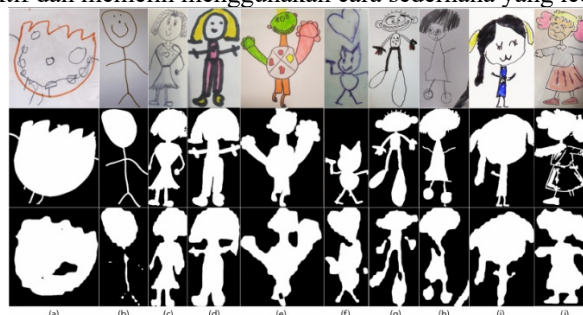
Langkah terakhir menentukan ukuran hasil kompres dari gambar asli dengan ukuran gambar hasil kompres. Seperti gambar dibawah ini.

```
import os
info = os.stat('jerawt6.webp')
print("sebelum: ",info.st_size/1024,"KB")
info = os.stat('download (3).png')
print("sesudah dikompresi: ",info.st_size/1024,"KB")
```

Gambar 15. Perbandingan hasil ukuran kompres dengan yang asli

Segmentasi Gambar

Segmentasi gambar merupakan pemisahan objek gambar dari latar belakang gambar dengan mengubah warna gambar asli ke warna abu abu, pada umumnya dalam mensegmentasi gambar banyak sekali yang menggunakan cara lain seperti menggunakan R-CNN seperti dilakukan oleh Smith et al dalam penelitiannya kurang efektif dan memelih menggunakan cara sederhana yang lebih evktif dan akurat.

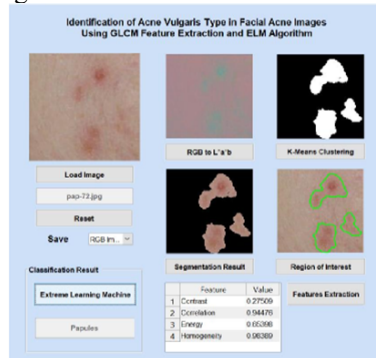


Gambar 16. Hasil Segmentasi Gambar menggunakan Model Klasik (Tengah) dan menggunakan model R-CNN (Bawah). Sumber : (Smith et al., 2023)

Segmentasi gambar juga disebut dengan segmentasi citra yaitu membagi beberapa kelompok piksel yang berada dalam satu kelompok besar Tingkat kemiripan lebih tinggi dibandingkan piksel kelompok kecil yang Tingkat kemiripanya rendah (Rachmat et al., 2022). Menurut IBM.com segmentasi gambar adalah teknik visi komputer yang mempartisi gambar digital ke dalam kelompok-kelompok piksel, segmen gambar, untuk menginformasikan deteksi objek dan tugas-tugas terkait. Dengan mengurai data visual gambar yang kompleks menjadi segmen berbentuk khusus, segmentasi gambar memungkinkan pemrosesan gambar yang lebih cepat dan lebih canggih. Sedangkan menurut Kamus Besar Bahasa Indonesia, Segmentasi adalah pemisahan atau pembagian suatu keseluruhan menjadi bagian-bagian atau kelompok-kelompok yang lebih kecil berdasarkan kriteria tertentu. Gambar menurut Kamus Besar Bahasa Indonesia adalah tiruan barang (orang, binatang, tumbuhan, dan sebagainya) yang dibuat dengan coretan pensil dan

sebagainya pada kertas dan sebagainya; lukisan. Selain itu, gambar juga bisa merujuk pada representasi visual dari sesuatu, yang dihasilkan pada suatu permukaan.

Metode Segmentasi Gambar merupakan metode sangat penting dalam penelitian, Metode ini sangat mudah dilakukan dalam pendeteksian seperti objek, wilayah dan pengelompokan lainnya dalam penelitian ini peneliti menggunakan metode ini untuk mendeteksi jera wat di wajah dengan membandingkan gambar asli dengan gambar yang sudah disegmentasi.



Gambar 17. Hasil Deteksi Jerawat menggunakan Matlab. Sumber : (Hasanah et al., 2022)

Mendeteksi jerawat sudah banyak yang melakukannya dengan berbagai macam metode ada yang menggunakan aplikasi web, Matlab (Gamabr 17), Yolo, Tensorflow, K-means Clustering, Viola Jones(Sajati, Indrianingsih, and Candra Wulan 2016). Dalam setiap jurnal penelitian yang peneliti baca dalam pemrosesan mensegmentasi gambar masih banyak menggunakan aplikasi lain, Seperti Matlab, Labeling dan aplikasi web lainnya. Sedangkan dalam penelitian ini peneliti melakukan pemrosesan mensegmentasi gambar menggunakan pemograman python di google colab. Setelah mengkompres gambar dilakukan maka langkah selanjutnya mensegmentasikan gambar menggunakan python di google colab seperti dibawah ini.

```
import numpy as np
import pandas as pd
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Gambar 18. Pemanggilan dataset di kaggle untuk mensegmentasikan gambar

Sebelum menjalankan program diatas langkah pertama yang harus dilakukan adalah membuka python di google colab lalu mengupload file gambar yang telah di kompres menggunakan metode algoritma K-means. Setelah data sudah siap langkah selanjutnya baru menjalan pemograman seperti gamabr di atas. Langkah selanjutnya jalankan pemograman seperti gamabr dibawah ini.

```
import cv2 as cv
from matplotlib import pyplot as plt
```

Gambar 19. Untuk menampilkan gambar segmentasi

Langkah berikutnya proses dan menampilkan gambar segmentasi untuk mendeteksi jerawat dengan pemograman python seperti di bawah ini.

```
img = cv.imread('/content/download (3).png',0)
edges = cv.Canny(img,50,100)
plt.subplot(121),plt.imshow(img,cmap = 'gray')
plt.title('Acne face'), plt.xticks([], plt.yticks([]))
plt.subplot(122),plt.imshow(edges,cmap = 'gray')
plt.title('Acne Detection'), plt.xticks([], plt.yticks([]))
plt.show()
```

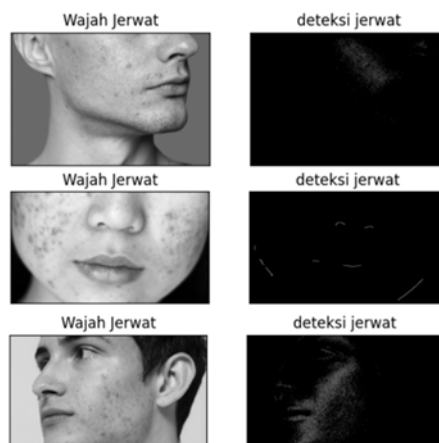
Gambar 20. Proses mendeteksi jerawat menggunakan segmentasi gambar

Jerawat Merupakan suatu penyakit kulit yang dialami anak remaja sampai dewasa yang biasa terjadi di kulit bagian Wajah, Lehe, Dada, Punggung yang disebabkan oleh pori-pori kulit tersumbat (Moncho-

Santonja et al., 2023). Jerawat kini merupakan masalah serius yang dialami oleh setiap anak karena mempengaruhi kurangnya penampilan, Kecantikan, dan ketampanan seseorang dalam beraktifitas sehari hari seperti bekerja, dan juga mencari jodoh. Dalam penelitian sudah banyak yang meneliti jerawat dengan berbagai metode yang sudah canggih dalam pemrosesan menurut peneliti, masih terlalu rumit untuk digunakan karena banyak tahap yang harus dilakukan, sedangkan dengan metode segmentasi gambar peneliti hanya menggunakan pemograman python yang sederhana dan mudah dilakan siapa saja untuk mendeteksi jerawat dengan mudah dan menghemat waktu. Walaupun dengan metode ini tidak mampu mengelompokkan jenis-jenis jerawat tetapi sudah bisa digunakan untuk mendeteksi jerawat dengan mencocokkan gambar dengan gambar aslinya.

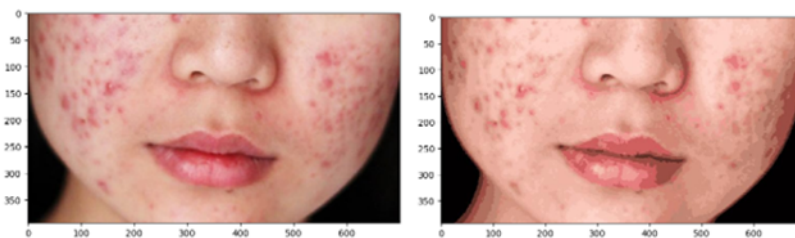
HASIL DAN PEMBAHASAN

Sebelum mensegmentasi gambar untuk mendeteksi jerawat yang pertama yang harus dilakukan adalah mengkompres gambar, agar gambar tersebut dapat di segmentasikan sebagaimana pelatihan percobaan bahwa gambar yang tidak dikompres dan langsung disegmentasi, jerawat tidak dapat terdeteksi seperti gambar dibawah ini.



Gambar 21. Hasil Jerawat tidak dapat di deteksi karena belum dikompres

Hasil dari gambar 21 sudah dapat disimpulkan bahwa jerawat tidak dapat dideteksi sebelum gambar di kompres terlebih dahulu, gambar sebelah kiri adalah gambar asli yang diubah warnanya dari asli ke abu-abu sedangkan gambar sebelah kanan adalah gambar hasil segmentasi yang digunakan untuk mendeteksi jerawat tapi tidak terdeteksi karna belum di kompres terlebih dahulu. Dan peneliti telah mengkompres gambar terlebih dahulu menggunakan algoritma K-means seperti contoh gambar di bawah ini.



Gambar 22. Hasil Kompres Gambar asli (kiri), gambar kompres (kanan)

Gambar sebelah kiri adalah gambar asli yang akan di kompres untuk mendeteksi jerawat menggunakan segmentasi gambar dan gambar sebelah kanan adalah gambar hasil kompres menggunakan algoritma K-means. Walaupun dalam penelitian ini peneliti mengalami kesulitan dalam pemograman karena tidak memiliki pengalaman soal pemograman python dalam mengkompres gambar yang seharusnya dalam mengekstrak gambar dari file gambar ukuran besar ke kecil justru terbalik dari file besar ke lebih besar lagi, walaupun dalam mengestrak gambar berhasil dalam perubahan warna asli ke warna kontras seperti gambar empat. seperti contoh gambar dibawah ini.

```
[ ] import os
info = os.stat('/content/wajah jerawat.jpg')
print("size of image before running k-mean algorithm:",info.st_size/1024,"KB")
info = os.stat('/content/habis kompres.png')
print("size of image after running k-mean algorithm: ",info.st_size/1024,"KB")
```

size of image before running k-mean algorithm: 26.5869140625 KB
 size of image after running k-mean algorithm: 213.4970703125 KB

Gambar 23. Hasil ukuran mengkompres gambar Menggunakan algoritma K-means



Gambar 24. Data hasil kompres gambar (bawah) dan data gambar asli (atas)

Setelah mengkompres gambar dilakukan langkah selanjutnya mensegmentasi gambar kompres ke warna abu-abu untuk mendeteksi jerawat berdasarkan kelompok berbentuk titik-titik dan lingkaran.

Dalam penelitian ini peneliti akan menunjukkan hasil pendeteksian jerawat menggunakan segmentasi gambar, dengan sampel data empat foto gambar berjerawat yang dimanah data yang digunakan adalah data sekunder yang diunduh dari google image sebagai bahan percobaan dalam penelitian.

Tabel 1. Table gambar mendeteksi jerawat menggunakan segmentasi gambar.

Picture	Kompres K means	Segmentation
		Acne Detection
		Acne Detection
		Acne Detection
		Acne Detection

Dari table diatas dapat disimpulkan bahwa mendeteksi jerawat menggunakan segmentasi gambar dapat dideteksi menggunakan segmentasi gambar dengan cara gambar asli harus di kompres terlebih dahulu baru gambar yang sudah di kompres di segmentasi. Setelah disegmentasi maka Langkah selanjutnya melakukan pencocokan gambar dengan gambar asli. Maka jerawat memiliki ciri ciri bentuk di segmentasi dengan berbentuk titik-titik atau lingkaran yang berkelompok.

KESIMPULAN

Penelitian ini dapat disimpulkan bahwa mendeteksi jerawat menggunakan segmentasi gambar dapat dilakukan jika gambar asli telah diestrak terlebih dahulu tanpa diestrak maka jerawat tidak dapat terdeteksi. Dengan metode ini maka mendeteksi jerawat bisa lebih cepat dan mudah dilakukan dengan menghemat waktu dan biaya.

UCAPAN TERIMA KASIH

Pertama saya mengucapkan terimakasih sebanyak-banyak kepada Universitas Amikom Yogyakarta tempat saya menimba ilmu dan terimakasih pula kepada dosen pembimbing saya yang selalu mensupport agar menyelesaikan kuliah ini, dan tak lupa pula kepada Prof. Indra Bastian yang selalu membantu saya, tanpa kalian saya bukan siapa siapa, dan terimakasih atas segalanya.

DAFTAR PUSTAKA

- Arifianto, July, and Izzati Muhimmah. 2021. "Aplikasi Web Pendeteksi Jerawat Pada Wajah Menggunakan Algoritma Deep Learning Dengan TensorFlow." *Journal Automata*: 21–29.
- Hasanah, Riyan Latifahul, and Muhamad Hasan. 2022. "Deteksi Lesi Acne Vulgaris Pada Citra Jerawat Wajah Menggunakan Metode K-Means Clustering." *Indonesian Journal on Software Engineering (IJSE)* 8(1): 46–51.
- Hasanah, Riyan Latifahul, Yan Rianto, and Dwiza Riana. 2022. "Identification of Acne Vulgaris Type in Facial Acne Images Using GLCM Feature Extraction and Extreme Learning Machine Algorithm." *Rekayasa* 15(2): 204–14.
- HASYDNA, NOVIA, and ROZZI KESUMA DINATA. 2020. "样本量估算-Machine Learning.Pdf"
- Moncho-Santonja, María, Silvia Aparisi-Navarro, Beatriz Defez, and Guillermo Peris-Fajarnés. 2023. "Segmentation of Acne Vulgaris Images Techniques: A Comparative and Technical Study." *Applied Sciences (Switzerland)* 13(10).
- Pinasty, Salsabila, Raden Bagus, and Fajriya Hakim. 2025. "Automatic Detection of Acne Types Using The YOLOv5 Method." *Indonesian Journal of Artificial Intelligence and Data Mining (IJAIMD)* 8(1): 236–48.
- Rachmat, Rachmat, Muhammad Yusuf, Irfan Abbas, and Muh. Fahmi Basmar. 2022. "Segmentasi Citra Menggunakan Algoritma K-Means Berbasis Particle Swarm Optimization." *Metik Jurnal* 6(2): 146–56.
- Sajati, Haruno, Yuliani Indrianingsih, and Puspa Ira Dewi Candra Wulan. 2016. "Deteksi Jerawat Pada Wajah Menggunakan Metode Viola Jones." *Compiler* 5(1).
- Smith, Harrison Jesse, Qingyuan Zheng, Yifei Li, Somya Jain, and Jessica K. Hodgins. 2023. "A Method for Animating Children's Drawings of the Human Figure." *ACM Transactions on Graphics* 42(3).
- Google, "Welcome to Colab!," google colab. [Online]. Available: <https://colab.research.google.com/#scrollTo=Wf5KrEb6vrkR>
- PYTHON, "What is Python? Executive Summary," PYTHON.ORG. [Online]. Available: <https://www.python.org/doc/essays/blurb/>
- IBM, "What is image segmentation?," IBM. [Online]. Available: <https://www.ibm.com/think/topics/image-segmentation>
- https://www.jurnas.com/images/posts/1/2023/2023-09_22/6d321a3f5a394a2949c162b0517863f4_1.jpeg
- <https://www.alodokter.com/jerawat-nodul-kenali-penyebab-dan-pengobatannya>