

Deteksi Perbedaan Gambar Dengan Menggunakan metode Structural Similarity Index Measurement (SSIM)

Khurin Rahmadini
Fakultas Sains dan Teknologi
Program Studi Teknik Informatika
Universitas Darussalam Gontor Kampus Putri
Sambirejo, Mantingan, Ngawi, Jawa Timur 63260
Khurinrahmadini@mhs.unida.gontor.ac.id

Abstract—Deteksi Perbedaan Citra dengan metode Structural Similarity Index measurement (SSIM) telah terbukti cukup efektif dalam melakukan segmentasi citra dimana citra dapat dieksekusi dengan konkrit namun dalam metode ini masih terdapat beberapa kekurangan diantaranya citra yang akan dieksekusi haruslah memiliki dimensi yang sama

Keywords—SSIM, Python, OpenCV

I. PENDAHULUAN

Pengolahan citra dewasa ini bukan lagi merupakan hal yang aneh, karena saat ini pengolahan citra telah digunakan dalam berbagai lini kehidupan masyarakat. Mulai dari hal pemindai sidik jari hingga menentukan penyakit tumor otak pun telah menggunakan citra digital. Pengolahan citra digital (*Digital Image Processing*) merupakan sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (yang berasal dari webcam). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer [1].

Pemanfaatan komputer untuk melakukan proses menemukan perbedaan antar gambar merupakan solusi yang ditawarkan, karena komputer pada dasarnya tidak memiliki rasa jenuh yang akan mempengaruhi hasil, selain itu dengan menggunakan komputer, pekerjaan akan dapat diselesaikan dengan cepat. Karena komputer akan mengolah data *digital*, maka masukan komputer berupa gambar harus merupakan gambar dalam format *digital*[2]. Ada banyak teknik yang digunakan namun pada penelitian kali ini penulis akan menggunakan metode Structural Similarity Index Measurement (SSIM).

Dalam pembahasan ini penulis mengkhususkan pemanfaatan metode dalam mendeteksi perbedaan gambar dalam sebuah ruangan yang mana dapat berfungsi disaat kita ingin merenovasi ulang sebuah ruangan

II. METODE

Dalam pembuatan penelitian ini penulis menggunakan metode *Structural Similarity Index Measurement* (SSIM) yang mana metode ini merupakan salah satu metode dalam mengukur kemiripan diantara 2 buah citra gambar dan berkorelasi kuat dengan kualitas persepsi Human Visual System (HVS) [3].

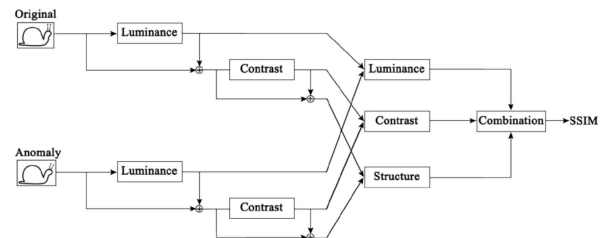


Fig. 1. SIIM Method[3].

A. Structural Similarity Index Measure (SSIM)

Algoritma SSIM (Structural Similarity Index Measure) merupakan salah satu metode dalam mengukur kemiripan diantara 2 buah citra gambar dan berkorelasi kuat dengan kualitas persepsi Human Visual System (HVS) [3]. Algoritma SSIM adalah metode membandingkan fitur struktural dari gambar, dan kualitas gambar dijelaskan oleh kesamaan struktural. Sebelum menggunakan SSIM untuk mengukur kualitas gambar, dibutuhkan gambar sebagai referensi. Kemudian meletakkan gambar eksperimen dan gambar referensi untuk melakukan pengukuran kesamaan struktural. Semakin tinggi kesamaan, semakin tinggi kualitas gambar, sedangkan semakin rendah kesamaan, semakin rendah juga kualitas gambar. Gambar adalah distribusi intensitas cahaya yang dipantulkan dari permukaan objek. Oleh karena itu, piksel gambar terkait dengan iluminasi dan refleksi dari permukaan objek. Namun demikian struktur dari objek tidak tergantung pada luminasi dan algoritma SSIM dapat memisahkan informasi struktur dari gambar[4].

B. Thresholding

Thresholding merupakan salah satu teknik segmentasi yang baik digunakan untuk citra dengan perbedaan nilai intensitas yang signifikan antara latar belakang dan objek utama (Katz,2000). Parameter kunci di dalam Thresholding merupakan pilihan dalam melakukan Threshold . Terdapat berbagai metode dalam memilih Threshold. Metode paling sederhana dilakukan dengan cara memilih nilai mean atau median. Pada dasarnya jika piksel objek lebih terang dibandingkan dengan background maka piksel objek tersebut juga lebih terang dari rata-ratanya. Pada gambar yang masih memiliki noise dengan background dan nilai objek, mean dan median akan bekerja maksimal dalam threshold. Dalam pendekatan yang lebih dalam, dapat pula dilakukan dengan cara membuat sebuah histogram dari intensitas citra piksel dan menggunakan valley point sebagai nilai threshold. Dengan melakukan pendekatan histogram memungkinkan adanya beberapa nilai rata – rata pada piksel background dan objek, tetapi nilai piksel tersebut mempunyai beberapa variasi nilai yang masih berada pada sekitar nilai rata – rata itu. Akan tetapi biasanya tidak selalu sesederhana itu dan banyak histogram dari citra yang mempunyai valley point yang tidak jelas [9]. Persamaan untuk menentukan nilai thresholding dapat dirumuskan sebagai berikut :

$$T = \frac{f_{maks} + f_{min}}{2}$$

Fig.2. rumus metode thresholding

III. HASIL

Metode *Singel Shot Scale-invariant Face Detector* (S3FD) yang terbilang cukup modern dibanding metode sebelumnya dalam *Face Detection* menjadikan metode ini banyak digunakan untuk mendeteksi citra, dan penggunaan tersebut bisa dibilang cukup mudah, dengan mengimplementasi beberapa konsep metode lain juga memudahkan penggunaannya. Untuk penerapannya dalam mendeteksi citra kita membutuhkan *OpenCV library* pada *Python* untuk mengolah citra yang akan dideteksi. Kali ini implementasi dilakukan dengan *Google Colab* menggunakan Bahasa *Python* dan library *OpenCV* serta beberapa tambahan *library* lain untuk melengkapi jalannya project ini, diantara langkah langkahnya adalah sebaga berikut :

A. Import Drive

Direktori yang digunakan dalam membuat project kali ini adalah *Google Drive*, karena pembuatannya yang *Online* membuat *Drive* butuh diimport kedalam *Google Colab*. Seluruh *dataset* yang akan digunakan dalam *project* ini disimpan dalam suatu folder di *Google Drive*, baik berupa Citra atau Tabel.

B. Import Library

Pada project ini dibutuhkan 2 buah library yaitu library open cv dan juga library imutils. *OpenCV* (*Open Source Computer Vision Library*) adalah salah satu *software* pustaka yang ditujukan untuk pengolahan citra dinamis

secara *real-time*, yang dibuat oleh *Intel*, dan sekarang didukung oleh *Willow Garage* dan *Itseez*. *OpenCV* bersama *Python* dimanfaatkan untuk mengolah suatu citra atau *image* atau video (tumpukan *frame/image*) sesuai dengan tujuan masing- masing yang melibatkan kamera untuk menangkap gambar lalu diolah di komputer [6]. Sedangkan library *Imutils* merupakan library untuk melakukan translation, rotation, resizing, and skeletonization.

```
# impor library yang diperlukan
from matplotlib import pyplot as plt
from skimage.measure import compare_ssim
import imutils
import cv2
```

Fig.3. library yang dibutuhkan

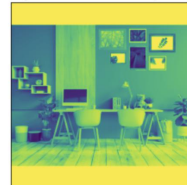
C. Mengubah Gambar Asli Menjadi Citra Keabuan

Citra grayscale merupakan citra yang menggunakan tingkatan warna abu-abu. Citra grayscale menyatakan nilai intensitas tiap piksel dengan nilai tunggal, dimana setiap piksel komponen merah, hijau, dan biru mempunyai intensitas yang sama. Citra grayscale lebih mudah diproses oleh karena hanya memiliki satu nilai intensitas pada setiap pixel. Intensitas nilai citra grayscale berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih (kadir, 2013) sehingga warna antaranya adalah abu-abu

```
# Konvert citra asli menjadi keabuan
grayA = cv2.cvtColor(imageA, cv2.COLOR_BGR2GRAY)
grayB = cv2.cvtColor(imageB, cv2.COLOR_BGR2GRAY)
```

```
[ ] plt.xticks([], plt.yticks([]))
plt.imshow(grayA)
plt.title('gambar awal,- ASLI - Grayscale')
plt.show()
```

gambar awal,- ASLI - Grayscale



```
[ ] plt.xticks([], plt.yticks([]))
plt.imshow(grayB)
plt.title('gambar kedua,- PERUBAHAN - Grayscale')
plt.show()
```

gambar kedua,- PERUBAHAN - Grayscale

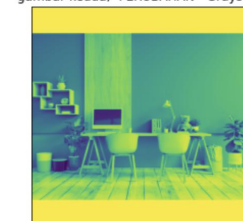


Fig.4. Citra greyscale

D. Menghitung Nilai SSIM

SSIM dikenal sebagai kualitas *metric* yang digunakan untuk mengukur kemiripan diantara 2 buah citra

dan dipercaya berkorelasi dengan kualitas persepsi *Human Visual System* (HVS) [21]. Model SSIM dibuat dengan memperhatikan 3 buah faktor yaitu *loss of correlation*, *luminance distortion* dan *contrast distortion*.

```
(score, diff) = compare_ssim(grayA, grayB, full=True)
diff = (diff * 255).astype("uint8")
print("SSIM: {} Rentang nilai -1 s/d 1 (nilai mendekati 1 berarti
antar citra hampir sempurna sama)".format(score))

SSIM: 0.9508825051847885 Rentang nilai -1 s/d 1 (nilai mendekati 1 ber
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: UserWa
This is separate from the ipykernel package so we can avoid doing ir

# threshold perbedaan citra, diikuti dengan pencarian kontur
# memperoleh area yang berbeda dari dua input citra
thresh = cv2.threshold(diff, 0, 255,
cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]
cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
cnts = imutils.grab_contours(cnts)
```

Fig.5. kontur

Kemudian melakukan perulangan pada kontur

```
# perulangan pada kontur
for c in cnts:
    # menghitung batas kotak dari kontur dan menggambar
    # batas kotak pada citra input untuk mewakili dua citra yang berbeda
    (x, y, w, h) = cv2.boundingRect(c)
    cv2.rectangle(imageA, (x, y), (x + w, y + h), (0, 0, 255), 2)
    cv2.rectangle(imageB, (x, y), (x + w, y + h), (0, 0, 255), 2)
```

Fig.6. Pengulangan pada Kontur

E. Menampilkan Kedua Hasil Citra

Pada tahap ini kita akan menampilkan kedua hasil citra yang telah melaewati tahap SSIM

```
# tampilkan citra
plt.xticks([], plt.yticks([]))
plt.imshow(cv2.cvtColor(imageA, cv2.COLOR_BGR2RGB))
plt.title('gambar awal,- ASLI')
plt.show()

plt.xticks([], plt.yticks([]))
plt.imshow(cv2.cvtColor(imageB, cv2.COLOR_BGR2RGB))
plt.title('gambar kedua,- PERUBAHAN')
plt.show()
```

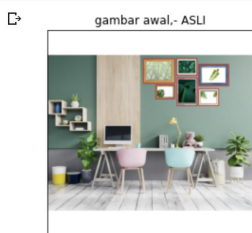
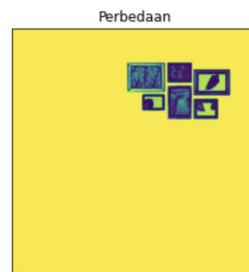


Fig.7. Hasil Citra

F. Mencari Perbedaan Dengan Metode Tresholding

Merupakan salah satu metode segmentasi citra yang memisahkan antara objek dengan background

```
plt.xticks([], plt.yticks([]))
plt.imshow(diff)
plt.title('Perbedaan')
plt.show()
```



```
plt.xticks([], plt.yticks([]))
plt.imshow(thresh)
plt.title('Threshold')
plt.show()
```

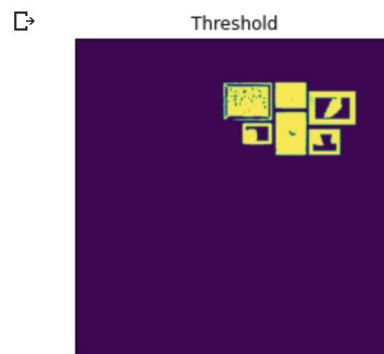


Fig.8. menemukan perbedaan Citra

G. Ringkasan

Keseluruhan dari tahap-tahap diatas merupakan keseluruhan cara-cara untuk menemukan perbedaan antara dua gambar. Dimulai dengan melakukan import drive lalu dilanjutkan dengan mengubah gambar menjadi citra keabuan atau greyscale agar gambar menjadi lebih sederhana kemudian melakukan penghitungan dengan metode SSIM dan yang terakhir yaitu mencari perbedaan gambar dengan menggunakan metode Tresholding

H. Hasil

Hasil dari percobaan tersebut sebagai berikut:

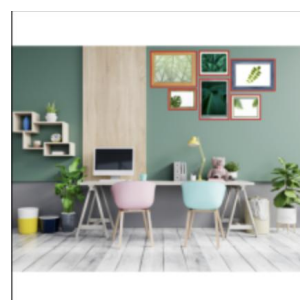


Fig.9. Citra awal

IV. KESIMPULAN

Penerapan pada penelitian perbedaan 2 gambar dengan menggunakan metode cukup baik karena memindai gambar dengan akurat namun kelemahan dari metode ini yaitu citra hanya dapat dideteksi apabila kedua citra memiliki dimensi yang sama.

REFERENCES

- [1] Kusumanto. RD, and Alan Novi Tompunu, "Pengolahan Citra Digital Untuk Mendeteksi Objek Menggunakan Pengolahan Warna Model Normalisasi RGB". Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2011 (Semantik 2011)
- [2] Harianto. Koko, "Penerapan Teknik Selisih Matriks untuk Menemukan Perbedaan Dua Buah Citra Digital". Jurnal SATIN-Sains Dan Teknologi Informasi, Vol.3, No.1, Juni 2014
- [3] Wang. Z, Bovik. AC, Sheikh. HR, Simoncelli. EP, "Image Quality Assessment: from error visibility to structural similarity", IEEE Transactions on Image processing, No.4, Vol.13, Hal. 600-612.
- [4] Chen. G, Shen. Y, Yao. F, Liu. P, "Region-based Moving Object Detection Using SSIM", 2015 4th international Conference on Computer Science and Network Technology (ICCSNT 2015), harbin, 19-20 Desember
- [5] Pramana. JC, "Implementasi Metode Thresholding dan Metode Regionprops Untuk Mendeteksi Marka Jalan Secara Live Video". Udinus Repo
- [6] J. Huang, V. Rathod, C. Sun, M.Zhu, A.Korattikaea, A. Fathi, I. Fischer, Z. Wonja, Y. Song, S. Guadar Rama, at al. Speed/accuracy trade-offs for modern convolutional object detectors. arXiv:1611.10012, 20
- [7] T. Mantoro, M. A. Ayu, and Suhendi, "Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods," Int. Conf. Multimed. Comput. Syst. -Proceedings, vol. 2018-May, pp. 1-5, 2018, doi: 10.1109/ICMCS.2018.8525935

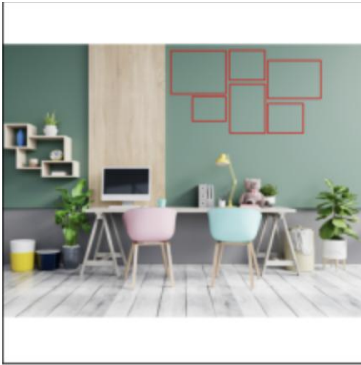


Fig.10. Mendeteksi Perbedaan Citra

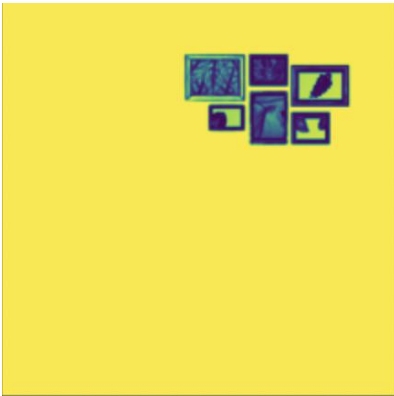


Fig.11. Menemukan Perbedaan Citra