

Aplikasi Pengenalan Wajah untuk Karakter Anime

Fransiskus Febryan Suryawan (13519124)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: franfebryan@gmail.com

Abstrak—Gambar karakter dengan gaya kartun Jepang atau biasa disebut anime saat ini banyak digunakan untuk berbagai keperluan, dengan contoh penggunaan paling banyak adalah sebagai persona dunia maya (seperti *vtuber/virtual youtuber*.) Aplikasi yang dibahas dalam makalah ini akan menggunakan pengolahan citra untuk mencoba mengenali wajah karakter anime tersebut untuk mengetahui lokasi *landmark* pada wajah. Data *landmark* kemudian dapat digunakan untuk melakukan proses lain, misalnya *morphing* dari wajah asli ke wajah karakter anime tertentu.

Kata Kunci—pengenalan wajah, anime, landmark wajah

I. PENDAHULUAN

Anime merupakan salah satu bentuk seni yang saat ini banyak digemari. Kata anime berasal dari kata bahasa Jepang アニメーション (*animēsyon*) yang diserap dari bahasa Inggris *animation*. Dalam KBBI sendiri, kata anime memiliki arti animasi atau kartun khas Jepang. Umumnya, anime merujuk pada animasi dan kartun yang memiliki gaya yang khas dan berasal dari Jepang. Namun, di Jepang, kata anime merujuk kepada sembarang animasi.

Gambar dengan gaya anime merujuk kepada gaya khusus yang digunakan dalam pembuatan gambar dalam anime. Namun, penggunaan frasa gaya anime tidak tertutup hanya dalam konteks anime saja, karena gambar dengan gaya anime saat ini banyak digunakan dalam berbagai sektor. Contoh dari penggunaan gambar dengan gaya anime adalah maskot untuk pemasaran.



Gambar. 1. Nanami Madobe, maskot Windows 7

Gambar dengan gaya anime umumnya digunakan dalam sektor hiburan. Salah satu bentuk penggunaan gaya anime

adalah dalam pembuatan persona di internet. Persona ini digunakan untuk melakukan aktivitas publik menggunakan identitas samaran, umumnya untuk membuat video hiburan. Saat ini, salah satu sektor yang banyak digemari adalah *vtuber* atau *virtual youtuber*, dimana seseorang menggunakan teknologi pengenalan wajah untuk menggerakkan gambar karakter bergaya anime dan memberikan kesan gambar menjadi hidup.



Gambar. 2. Kobo Kanaeru, *vtuber* asal Indonesia yang berada dalam naungan agensi Hololive

Teknologi pengolahan dan interpretasi citra yang ada saat ini mampu mengenali wajah serta *landmark* wajah menggunakan citra wajah sebagai masukan. Namun, pengenalan wajah umumnya hanya digunakan pada wajah manusia. Dalam makalah ini, akan dibahas mengenai pengenalan *landmark* wajah pada gambar karakter bergaya anime.

II. STUDI LITERATUR

A. Citra

Menurut [1], sebuah citra dapat didefinisikan sebagai fungsi dwimatra $f(x, y)$ dengan x, y adalah koordinat spasial dan keluaran fungsi f adalah intensitas atau tingkat keabuan pada titik x, y . Umumnya, citra digital memiliki koordinat x, y yang diskrit. Citra diskrit pada ranah digital didapatkan dengan melakukan penerokan terhadap citra pada titik-titik tertentu, lalu melakukan kuantisasi dari nilai-nilai tingkat keabuan yang diterok. Hasil penerokan memberikan citra yang dapat direpresentasikan sebagai matriks dengan M baris dan N kolom seperti pada persamaan 1.

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix} \quad (1)$$

Untuk sembarang citra, proses kuantisasi memetakan tingkat keabuan pada sejumlah L tingkat keabuan yang mungkin ada. Tingkat diskrit pada L dapat diasumsikan merupakan bilangan bulat yang tersebar merata dalam interval $[0, L-1]$. Umumnya, L merupakan suatu pangkat bilangan bulat dari 2: $L = 2^k$.

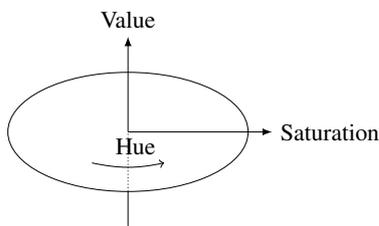
Citra yang berwarna dapat direpresentasikan dengan menyimpan tingkat keabuan untuk tiga warna dasar, yakni merah (R), hijau (G), dan biru (B).

B. Warna

Warna yang dapat direspon oleh mata manusia memiliki panjang gelombang antara 400 nm hingga 700 nm. Terdapat tiga warna yang memiliki kombinasi rentang terbesar, yakni merah, hijau, dan biru. Ketiganya seringkali disebut sebagai warna pokok. Warna-warna lain dapat dihasilkan dengan menggabungkan ketiga warna pokok dengan perbandingan tertentu.

Terdapat beberapa model warna yang terdefinisi. Model warna yang umumnya digunakan untuk penyimpanan dan pengolahan antara lain model warna RGB, CMYK, serta YCbCr. Model warna RGB menggunakan warna pokok seperti yang disebutkan sebelumnya. Model warna CMYK menggunakan warna sekunder yang dihasilkan dari penggabungan merah, hijau, dan biru, dengan tambahan warna hitam sebagai key. Model warna YCbCr menggunakan kecerahan serta kroma pada warna biru dan merah.

Selain model warna untuk penyimpanan dan pengolahan, terdapat pula model warna yang merepresentasikan pengalaman indera manusia secara lebih baik. Model warna tersebut adalah HSV, yang memisahkan warna, kecerahan, dan saturasi. Dalam model warna HSV, warna dikodekan sebagai *hue*, *saturation*, dan *value*. Nilai *hue* merepresentasikan warna dalam rentang $[0, 2\pi]$. Nilai *saturation* merepresentasikan kepekatan warna dalam rentang $[0, 1]$. Nilai *value* merepresentasikan kecerahan warna dalam rentang $[0, 1]$. Ilustrasi dari model warna HSV dapat dilihat pada gambar 3.



Gambar. 3. Ilustrasi model warna HSV.

C. Deteksi Tepi

Disebutkan dalam [2], tepi merupakan salah satu fitur yang ada dalam citra. Tepi didefinisikan sebagai perubahan nilai keabuan yang besar dalam jarak spasial yang pendek. Terdapat empat macam tepi: tepi curam, tepi landai, tepi garis, dan tepi atap. Pendeteksian tepi berguna untuk meningkatkan penampakan garis batas objek, serta mengekstraksi representasi garis-garis dalam citra.

Pendeteksian tepi dapat dilakukan dengan menggunakan beberapa operator. Operator yang paling sederhana adalah operator gradien, yakni operator yang menghitung kemiringan dari fungsi. Variasi dari operator ini adalah operator gradien dengan selisih-terpusat (*center-difference*).

Operator lain yang mungkin digunakan dalam pendeteksian citra adalah operator Laplace atau turunan kedua. Operator Laplace menggunakan turunan kedua untuk mencari perubahan tanda gradien, yakni dengan mencari persilangan nol (*zero-crossing*). Operator Laplace didapatkan dengan mengambil gradien orde kedua dari $f(x, y)$, atau menurunkan $f(x, y)$ secara diskrit sebanyak dua kali. Operator Laplacian dapat dilakukan dengan melakukan konvolusi menggunakan kernel pada persamaan 2.

$$D_{xy}^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2)$$

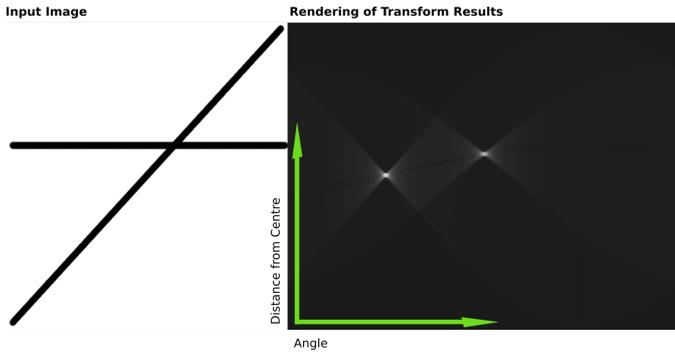
Derau dapat dideteksi sebagai tepi akibat perubahan nilai keabuan yang cepat. Oleh karena itu, perlu ditapis terlebih dahulu. Salah satu penapis yang mungkin digunakan adalah penapis Gaussian, yang melembutkan citra. Dalam kasus operasi Laplace, penapis Gaussian dapat digabungkan dengan operasinya. Oleh karena itu, perhitungan tepi menggunakan operasi Laplace dapat dilakukan lebih efisien. Operator Laplace yang dilengkapi dengan Gaussian umumnya disebut sebagai operator tersendiri, yakni Laplacian of Gaussian (LoG). Kernel untuk LoG didapatkan dengan menggunakan persamaan 3.

$$\text{LoG}(u, v) = \frac{1}{\pi\sigma^4} \left(\frac{u^2 + v^2}{2\sigma^2} - 1 \right) e^{-\frac{u^2 + v^2}{2\sigma^2}} \quad (3)$$

Selain kedua operasi tersebut, terdapat operasi Canny yang dapat digunakan untuk mendeteksi tepi. Tepi yang dideteksi oleh operator Canny akan memiliki lebar 1 pixel, berbeda dengan operasi-operasi lainnya. Operator Canny mendapatkan tepi dengan langkah-langkah sebagai berikut:

- 1) Haluskan citra menggunakan penapis Gaussian
- 2) Hitung gradien serta arahnya menggunakan operator deteksi tepi lainnya
- 3) Ambangkan nilai tepi yang didapat sebelumnya

Pengambangan dalam operator Canny menggunakan dua nilai ambang, $T1$ dan $T2$ untuk mengklasifikasi tepi menjadi tepi kuat dan tepi lemah. Jika magnitudo tepi melebihi $T2$, maka tepi tersebut diklasifikasikan menjadi tepi kuat. Apabila magnitudo tepi berada antara $T1$ dan $T2$, maka tepi tersebut diklasifikasikan sebagai tepi lemah. Di luar kasus tersebut, maka dianggap sebagai bukan tepi. Hasil dari operator Canny



Gambar. 4. Contoh hasil transformasi Hough untuk citra dengan 2 garis tebal. Diambil dari Wikipedia

merupakan tepi kuat serta tepi lemah yang berhubungan dengan tepi kuat.

D. Kontur

Menurut [3], kontur merupakan serangkaian tepi yang membentuk batas daerah tertentu. Kontur dapat direpresentasikan sebagai senarai tepi (*edge list*) atau berupa kurva. Kode rantai dapat digunakan untuk mengkodekan senarai tepi.

Pencocokan kurva dalam tepi dapat dilakukan dengan melakukan interpolasi dan penghampiran. Salah satu metode untuk melakukan penghampiran kurva pada tepi adalah transformasi Hough. Teknik tersebut dapat menghampiri kurva parametrik dalam tepi.

Transformasi Hough menggunakan bentuk parametrik kurva untuk melakukan penghampiran. Kemudian, mekanisme yang dilakukan untuk mengestimasi nilai parameter adalah dengan pemungutan suara untuk setiap kombinasi parameter. Setiap titik tepi memberikan suara untuk kombinasi parameter tertentu. Parameterisasi untuk garis lurus dapat dilakukan dengan mengubah persamaan garis lurus $y = mx + c$ ke dalam ruang parameter $m - c$: $c = y - mx$. Namun, kelemahan ruang parameter ini adalah adanya nilai gradien m yang tidak terdefinisi untuk garis tegak lurus. Oleh karena itu, alternatif parameterisasi yang mungkin dilakukan adalah dengan mengubah persamaan garis ke bentuk normal Hesse, yakni $r = x \cos \theta + y \sin \theta$. Transformasi Hough memberikan nilai kemungkinan adanya garis dengan parameter dalam ruang parameter. Garis yang didekati dapat diambil berdasarkan nilai ambang untuk jumlah suara tertentu. Visualisasi hasil transformasi Hough untuk citra tertentu dapat dilihat pada gambar 4.

E. Segmentasi

Suatu citra dapat disegmentasi menjadi beberapa bagian, berdasarkan kriteria tertentu. Segmentasi dapat berguna untuk memisahkan objek yang ingin diproses dengan hal di sekitarnya. Umumnya teknik segmentasi citra dapat dikelompokkan menjadi dua pendekatan, yakni berbasis diskontinuitas serta berbasis kesamaan.

Pendeteksian segmen menggunakan kesamaan salah satunya dapat dilakukan dengan pengambangan. Pada citra orang atau

karakter anime, umumnya terdapat warna kulit yang diterima agar dapat didefinisikan sebagai karakter. Oleh karena itu, segmentasi dapat dilakukan dengan melakukan pengambangan pada model warna HSV.

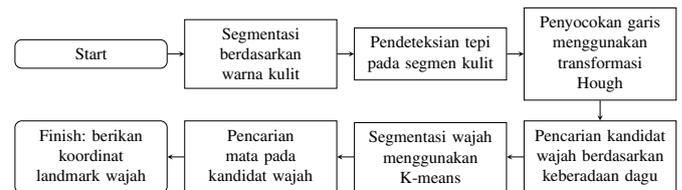
Selain menggunakan pengambangan, segmen dapat dibangun menggunakan algoritma *K-Means*. Algoritma *K-Means* merupakan algoritma yang dapat mengelompokkan data menjadi k kelompok berdasarkan ukuran kemiripan yang diberikan. Penggunaan algoritma *K-Means* adalah untuk mensegmentasi citra menjadi beberapa segmen warna.

III. STUDI TERKAIT

Dalam [4], dideskripsikan cara untuk mendeteksi wajah karakter kartun menggunakan teknik-teknik pengolahan citra. Dalam studi tersebut, digunakan nilai *hue* tertentu untuk mendeteksi wajah berdasarkan warna kulit. Nilai rentang *hue* yang digunakan adalah $[0, 40]$ dalam derajat. Selain itu, faktor *value* juga digunakan dalam untuk mendeteksi warna kulit. Nilai rentang yang digunakan adalah $[0.75, 1]$.

IV. IMPLEMENTASI SOLUSI

Solusi dibuat menggunakan bahasa pemrograman Python versi 3.9. Pustaka OpenCV versi 4.6 serta NumPy digunakan untuk membantu melakukan berbagai operasi pada citra secara efisien. Alur umum solusi dapat dilihat pada gambar 5.



Gambar. 5. Diagram alur solusi

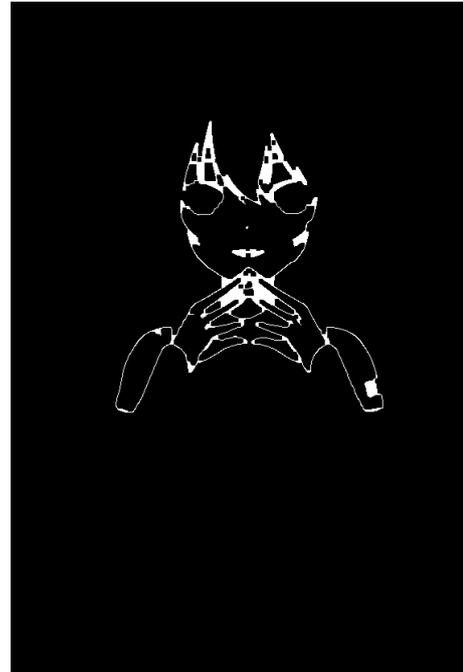
Masukan dari program adalah sebuah citra yang berisi wajah karakter anime. Keluaran dari program adalah titik-titik yang membentuk kontur wajah serta titik *landmark* berupa dua buah titik mata dan satu buah titik dagu.

Proses pertama yang dilakukan oleh program adalah melakukan segmentasi gambar untuk mengekstraksi bagian yang berisi warna kulit. Berdasarkan penelitian [4], nilai *hue* yang cocok menjadi batas ambang adalah 40. Namun, berdasarkan eksperimen, nilai ambang 60 menghasilkan segmentasi yang lebih baik. Selain itu, hasil segmentasi menghasilkan pixel-pixel putih yang tidak diinginkan, sehingga pixel-pixel putih dibuang. Hasil segmentasi kemudian dideteksi tepinya. Dalam solusi yang dibuat, Segmentasi berdasarkan warna digunakan untuk mengurangi tepi-tepi yang tidak digunakan. Citra yang digunakan sebagai contoh dapat dilihat pada gambar 6. Visualisasi hasil segmentasi kulit dapat dilihat pada gambar 7. Visualisasi hasil pendeteksian tepi dapat dilihat pada gambar 8.

Selanjutnya, untuk setiap daerah atau kontur tertutup, dilakukan pencocokan menggunakan transformasi Hough. Pencocokan garis dilakukan untuk mencari garis-garis yang membangun wajah. Berdasarkan observasi, garis yang membangun



Gambar. 6. Citra Hatsune Miku yang digunakan untuk percobaan



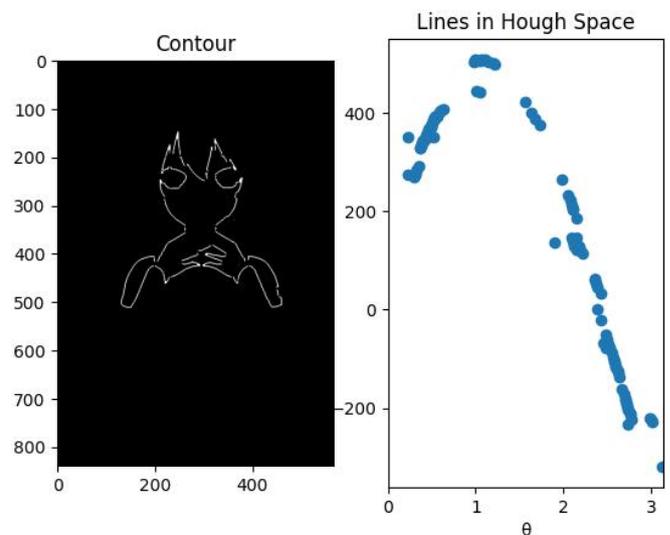
Gambar. 8. Visualisasi pendeteksian tepi pada segmen kulit



Gambar. 7. Visualisasi ekstraksi warna kulit

wajah pada karakter anime relatif lurus. Kemudian, terdapat garis yang bertemu pada titik dagu. Oleh karena itu, pendeteksian dagu dilakukan dengan mencari dua buah pasangan titik yang bertemu pada titik di dalam segmen pada rentang sudut tertentu. Dari eksperimen, didapatkan rentang sudut yang cukup optimal untuk wajah yang menghadap depan adalah

[50, 100] dalam derajat. Namun, rentang tersebut mungkin berubah tergantung pada gaya penggambaran spesifik yang digunakan. Visualisasi garis hasil transformasi Hough pada salah satu kontur dapat dilihat pada gambar 9.



Gambar. 9. Visualisasi transformasi Hough untuk kontur

Setelah mendapatkan kontur yang mungkin berisi wajah beserta titik dagu, mata akan dicari dalam kontur tersebut. Pencarian mata dilakukan dengan pertama-tama melakukan segmentasi berdasarkan warna pada bagian wajah menggunakan algoritma *K-Means*. Berdasarkan eksperimen, nilai K

yang optimal adalah 3, karena umumnya dalam wilayah wajah, tidak banyak warna selain warna kulit dan warna rambut. Visualisasi hasil segmentasi menggunakan *K-Means* dapat dilihat pada gambar 10. Dari hasil segmentasi, diambil segmen dengan jumlah pixel paling sedikit. Visualisasi segmen dengan jumlah pixel paling sedikit dapat dilihat pada gambar 11.



Gambar 10. Visualisasi hasil *K-Means* dengan $k = 3$



Gambar 11. Visualisasi segmen yang mungkin mencakup mata

Berdasarkan observasi, umumnya mata pada karakter anime memiliki *outline* yang umumnya berwarna gelap, serta simetris antara kanan dan kiri. Oleh karena itu, akan dicari komponen dengan kontur yang memiliki bentuk yang mirip. Selain memiliki bentuk yang mirip, umumnya mata juga memiliki warna yang sama, maka akan dicari segmen yang memiliki warna yang mirip. Dalam eksperimen, didapatkan nilai kemiripan berada di bawah ambang (45, 65, 40) dalam model warna HSV. Visualisasi hasil pasangan mata yang didapatkan dapat dilihat pada gambar 12.

Setelah mata didapatkan, maka proses akhir adalah dengan mengambil titik tengah dari *bounding box* segmen mata, untuk mengaproksimasi posisi mata. Visualisasi hasil akhir dari solusi dapat dilihat pada gambar 13. Visualisasi hasil akhir memberikan *bounding box* wajah yang ditemukan dalam



Gambar 12. Visualisasi pasangan mata yang didapatkan

persegi berwarna hijau dan *landmark* ditandai dengan titik berwarna merah.



Gambar 13. Visualisasi titik-titik *landmark*

V. PENGUJIAN

Citra lain digunakan untuk menguji kebenaran solusi yang diberikan. Citra yang akan digunakan dapat dilihat pada gambar 14. Citra tambahan tidak diproses terlebih dahulu sebelumnya, dan hasilnya dapat dilihat pada gambar 15.

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Hasil dari pengujian menunjukkan masih banyak kesalahan pendeteksian wajah. Dari titik-titik yang didapatkan, kemungkinan kesalahan terjadi pada saat pendeteksian dagu. Karena posisi dagu yang tidak diketahui, maka sulit untuk menentukan apakah dua buah garis yang bersilangan membangun dagu atau tidak. Hal tersebut mempengaruhi pendeteksian mata yang dilakukan, karena pendeteksian mata didasarkan pada kemiripan jarak antara mata dengan dagu.

B. Saran

Penggunaan kaskas yang lebih canggih seperti *Haar Cascade Classifier* dapat mempercepat proses pencarian *landmark* wajah maupun pendeteksian wajah itu sendiri. Menggunakan



Gambar. 14. Citra uji tambahan



Gambar. 15. Hasil pengujian pada citra uji tambahan

cascading classifier, pencarian pola dapat dilakukan dengan lebih efektif dibandingkan *brute-force*.

PRANALA TERKAIT

Repository GitHub: <https://github.com/suggoitanoshi/if4073-pengenalan-anime>

UCAPAN TERIMA KASIH

Syukur kepada Tuhan Yang Maha Esa atas karunia-Nya sehingga penulis dapat menyelesaikan makalah ini tanpa kesulitan maupun tantangan yang terlalu berat. Terima kasih yang teramat banyak kepada Bapak Rinaldi selaku dosen pengampu mata kuliah IF4073 Interpretasi dan Pengolahan Citra, sehingga penulis dapat belajar lebih lanjut mengenai citra dan pengolahannya. Ucapan terima kasih juga diberikan kepada keluarga dan teman-teman yang senantiasa membantu dan memberikan semangat kepada penulis sehingga penulis tidak menyerah dalam menyelesaikan perkuliahan di semester ini.

REFERENSI

- [1] R. C. Gonzalez, *Digital image processing*. Pearson education india, 2009.
- [2] R. Munir, "18 - pendeteksian tepi (edge detection) (bagian 1)," 2022.
- [3] R. Munir, "21 - kontur," 2022.
- [4] K. Takayama, H. Johan, and T. Nishita, "Face detection and face recognition of cartoon characters using feature extraction," in *Image, Electronics and Visual Computing Workshop*, p. 48, 2012.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 16 Desember 2022

Fransiskus Febryan Suryawan
13519124