Pewarnaan Kuku Jari Tangan Secara Digital untuk Nail Art

Mellisa Irawan / 13619063
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): mellisairawan45@gmail.com

***Abstrak*—Pada makalah ini akan dilakukan implementasi program Python untuk memberikan gambaran hasil aplikasi variasi warna cat kuku dari penyedia jasa *nail art.* Program yang dibuat akan menggunakan teknik-teknik pengolahan citra meliputi pendeteksian tepi, penghalusan citra, segmentasi citra, pembuatan kontur, hingga penjumlahan citra dengan pembobotan. Dari hasil proses pengolahan citra di makalah ini, program mampu memberikan warna pada citra masukkan berupa kuku pengguna dengan warna/pola cat kuku yang diambil dari citra masukkan lainnya.**

Kata kunci—nail art, pendeteksian tepi, segmentasi citra

I. PENDAHULUAN

Nail art adalah seni untuk menghias kuku pada umumnya dengan mengaplikasikan cat berwarna khusus pada kuku. Selain mempercantik kuku, *nail art* juga bisa menutupi kekurangan kuku misalnya menutupi warna kuku yang tidak rata. Banyak salon-salon saat ini yang secara khusus menyediakan jasa untuk memberikan *nail art* kepada pelanggannya. Untuk menentukan warna yang cocok pada kuku, biasanya seseorang akan mencoba satu per satu warna-warna yang disediakan termasuk menggunakan produk alcohol khusus untuk menghapus *nail art* dengan cepat. Proses ini membuat rugi baik bagi penyedia jasa *nail art* maupun pelanggan pengguna jasa *nail art* itu sendiri. Penyedia jasa *nail art* harus menghabiskan sejumlah cat kuku hanya digunakan untuk uji coba kecocokkan oleh pelanggan dan juga menyediakan produk alkohol untuk menghapus cat kuku dengan cepat. Bagi pelanggan, penggunaan alkohol yang terlalu banyak untuk menghapus cat kuku dapat merusak kualitas kuku. Selain itu, pelanggan perlu secara langsung datang ke penyedia jasa *nail art* untuk mengetahui variasi warna cat apa saja yang tersedia.

Suatu solusi untuk menyelesaikan masalah ini adalah dengan membuat sebuah program yang dapat diakses pengguna untuk memberikan gambar dimana pengguna dapat melihat hasil aplikasi variasi warna cat kuku penyedia jasa *nail art.* Dengan program ini pengguna dapat mendapat gambaran hasil penggunaan warna cat kuku tertentu serta dapat mencoba program ini tanpa harus datang ke toko *nail art.*

Program ini dapat dibuat menggunakan teknik-teknik pengolahan citra. Pertama-tama program menerima gambar masukkan berupa gambar tangan pengguna. Dari gambar tangan ini, program akan melakukan deteksi tepi untuk membuat penapis yang mendefinisikan lokasi kuku jari dan memisahkannya dari keseluruhan gambar tangan untuk dibuat sebuah *mask*. Kemudian program juga menerima satu gambar masukkan berupa warna/pola cat kuku yang disediakan oleh penyedia jasa *nail art.* Gambar warna/pola cat kuku ini kemudian diambil menggunakan *mask* dari kuku yang sudah dibuat kemudian dilakukan *overlay* gambar ini ke gambar masukkan tangan pengguna. Keluaran program adalah gambar tangan pengguna dimana cat telah diaplikasikan pada kuku.

Gambar 1Variasi cat kuku yang terdiri dari banyak warna

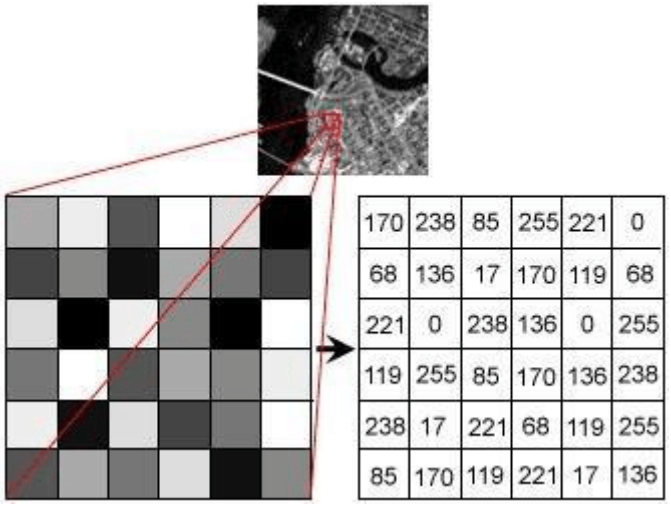
II. LANDASAN TEORI

A. Citra Digital

Citra dapat digolongkan menjadi dua jenis, yaitu citra analog dan citra digital berdasarkan bentuk sinyal yang menyusun citra tersebut. Citra analog adalah citra yang dihasilkan dari alat akuisisi citra analog seperti mata manusia. Citra tipe ini

memiliki kualitas resolusi yang sangat bagus namun tidak bisa disimpan maupun diolah oleh komputer. Oleh karena itu, citra analog perlu diubah ke citra digital dengan cara penerokan *(sampling)* untuk menyatakan ukuran citra dan kuantisasi untuk menentukan berapa nilai *pixel*.

Citra digital merepresentasikan fungsi intensitas cahaya dalam bentuk diskrit yang dibentuk pada matriks dua dimensi M x N dimana M x N menyatakan resolusi citra. Sebuah citra digital terdiri atas *pixel-pixel (picture element)* dimana setiap *pixel* menempati satu koordinat *(x, y)* yang menunjukkan lokasi *pixel* dalam matriks dua dimensi dan citra digital juga memiliki amplitudo *f(x, y)* yang menunjukkan nilai intensitas warna citra.



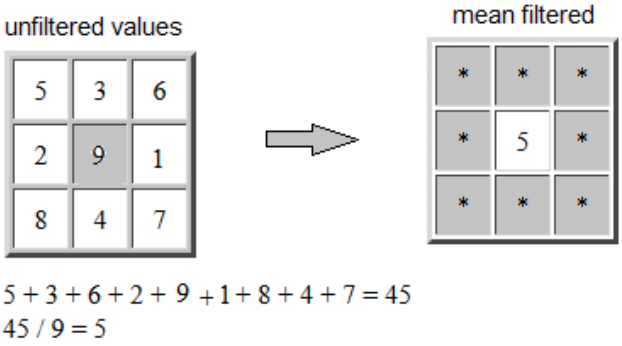
Gambar 2 Representasi citra digital grayscale yang terdiri atas pixel-pixel dengan niai 0 sampai 255 (8-bit)

Gambar 3 Pemecahan citra RGB menjadi tiga kanal warna

Berdasarkan kombinasi warna, terdapat tiga jenis citra yaitu citra berwarna, citra grayscale, dan citra biner. Citra berwarna memiliki tiga kanal warna yaitu *red (R), green (G),* dan *blue (B).* Setiap kanal pada dasarnya adalah citra *graylevel* dengan panjang 8-bit namun nilai 255 untuk masing-masing kanal adalah merah, hijau, dan biru. Citra *grayscale* adalah citra yang umumnya 1 *pixel* bernilai 8 bit yang merepresentasikan derajat keabuan sedangkan citra biner hanya terdiri atas *pixel* hitam dan *pixel* putih.

B. Penghalusan Citra/Blurring

Pelembutan citra dilakukan untuk mengurangi derau pada citra. Selain itu, pelembutan citra juga dilakukan untuk mengurangi jumlah tepi pada gambar sehingga pendeteksian tepi yang dilakukan dapat lebih akurat mengindentifikasi fitur objek. Pelembutan citra pada umumnya dilakukan dengan mengganti nilai *pixel* dengan nilai rata-rata atau nilai median dari *pixel* tersebut dengan *pixel-pixel* tetangganya ataupun meng. Untuk melakukan pelembutan, dilakukan operasi konvolusi dengan penapis rerata atau penapis median. Penapis rerata adalah sebuah matriks dua dimensi dimana jumlah setiap nilai dari seluruh elemen matriksnya adalah satu sedangkan penapis median hanya mengambil nilai *pixel-pixel* citra kemudian mencari nilai mediannya. Hasil dari proses pelembutan menimbulkan efek *blurring* pada citra.



Gambar 4 Contoh proses kalkulasi untuk melakukan pelembutan citra dengan penapis rerata

Gambar 5 Hasil penghalusan citra dengan penapis rerata dan penapis median

C. Pendeteksian Tepi

Pendeteksian tepi dilakukan untuk meninkatkan penampakan garis-garis batas yang membentuk objek-objek di dalam citra. Tepi *(edge)* adalah perubahan nilai intensitas nilai keabuan yang mendadak dalam jarak yang singkat yang

biasanya terdapat pada batas antara dua daerah yang berbeda. Karena pada dasarnya tepi adalah perubahan intensitas pada gambar, maka tepi dapat dideteksi dengan operator yang menghitung turunan pertama *(gradient)*.

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

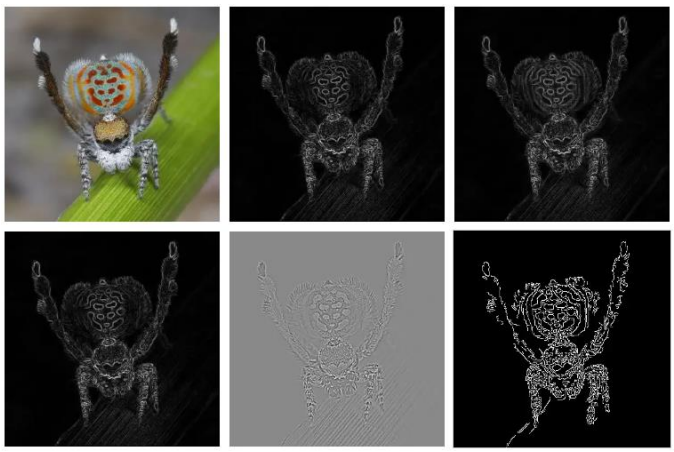
Gambar 6 Dua contoh matriks operator Laplacian

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Gambar 7 Matriks operator Sobel

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Gambar 8 Matriks operator Prewitt

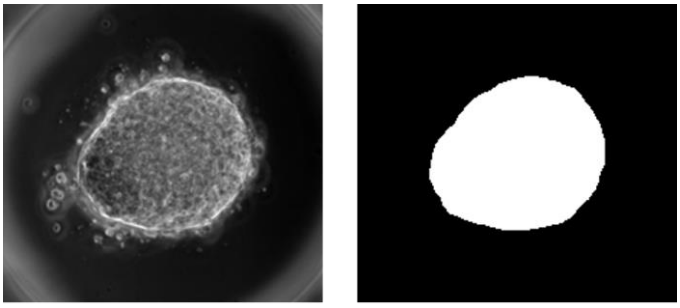


Gambar 9 Contoh hasil pendeteksian citra, dari kiri atas ke kanan bawah: citra asli, citra hasil operator Sobel, citra RGB hasil operator Sobel, citra hasil operator Prewitt, citra hasil operator Laplacian, dan citra hasil operator Canny.

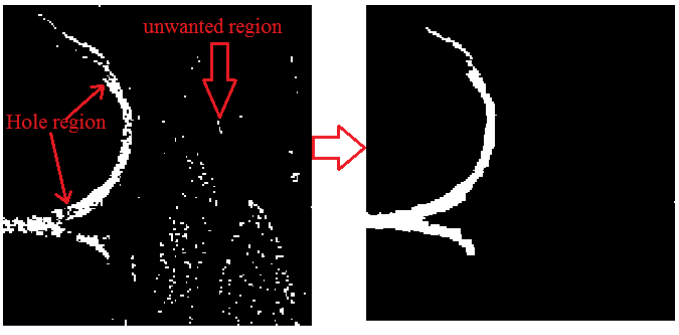
Selain itu, pendekatan operator turunan kedua *(Laplacian)* dapat digunakan untuk mendeteksi lokasi tepi dengan lebih akurat karena dengan turunan kedua yang dapat mendeteksi persilangan nol yang dimiliki tepi yang curam. operator Laplacian ini juga memiliki modifikasi menjadi operator Laplacian of Gaussian dimana dilakukan penghalusan citra dengan fungsi Gauss terlebih dahulu sebelum menggunakan operator Laplacian. Adapun operator-operator lain yang juga dapat digunakan untuk mendeteksi tepi: operator Sobel, operator Roberts, operator Prewitt, dan operator Canny. Operator Sobel, Roberts dan Prewitt masing-masing memiliki matriks *mask* konvolusi dengan nilai elemen matriks yang berbeda sedangkan operator Canny mendeteksi tepi dengan cara menghaluskan citra dengan penapis Gaussian, menghitung gradien dan arah gradien setiap *pixel* dengan salah satu operator Sobel, Roberts atau Prewitt, dan kemudian melakukan pengambangan.

D. Segmentasi Citra

Segmentasi citra dilakukan untuk mempartisi citra menjadi kelompok-kelompok *pixel* yang saling terhubung berupa region-region (objek), struktur linier (garis, kurva), maupun benntuk dua dimensi (lingkaran, elips, kotak, dll). Salah satu metode segmentasi adalah dengan cara mempartisi objek-objek pada citra dengan mendeteksi tepi objek pada citra *grayscale* kemudian melakukan pengambangan sehingga menghasilkan citra biner. Citra biner ini akan digunakan sebagai *mask* untuk memisahkan objek dengan latar belakang citra semula. Salah satu metode segmentasi lainnya adalah dengan menggunakan penapis luas untuk mengeliminasi daerah-daerah gangguan yang berukuran kecil.



Gambar 10 Contoh segmentasi citra untuk menghasilkan mask berupa citra biner



Gambar 11 Contoh penggunaan penapis luas untuk mengeliminasi daerah-daerah gangguan yang berukuran kecil

E. Citra Transparan

Citra transparan berwarna pada dasarnya menggunakan model warna RGB namun terdapat tambahan kanal warna *alpha (α)* yang menggambarkan tingkat transparansi gambar. Maka dari itu, citra transparan memiliki format Pada pengaplikasian di kasus pewarnaan cat kuku ini, cat kuku perlu diberi transparansi supaya terlihat natural pada gambar tangan. Dengan Python, penggabungan kedua gambar dapat diberi transparansi dengan menggunakan fungsi *cv2.addWeighted*. Dengan fungsi ini, dilakukan operasi penjumlahan pada kedua citra namun keduanya diberi pembebanan supaya dapat *pixel-pixel* dari kedua gambar dapat muncul bersamaan. Fungsi ini melakukan penambahan pada gambar dengan persamaan:

$$G(x) = (1 - \alpha) \cdot f_1(x) + \alpha \cdot f_2(x)$$

dimana α adalah nilai transparansi dengan nilai 0 hingga 1, $f_1(x)$ adalah nilai *pixel* pada lokasi *x* dari gambar 1, dan $f_1(x)$ adalah amplitudo *pixel* pada lokasi *x* dari gambar 2. Dengan demikian, citra yang dihasilkan tetap dalam format RGB.

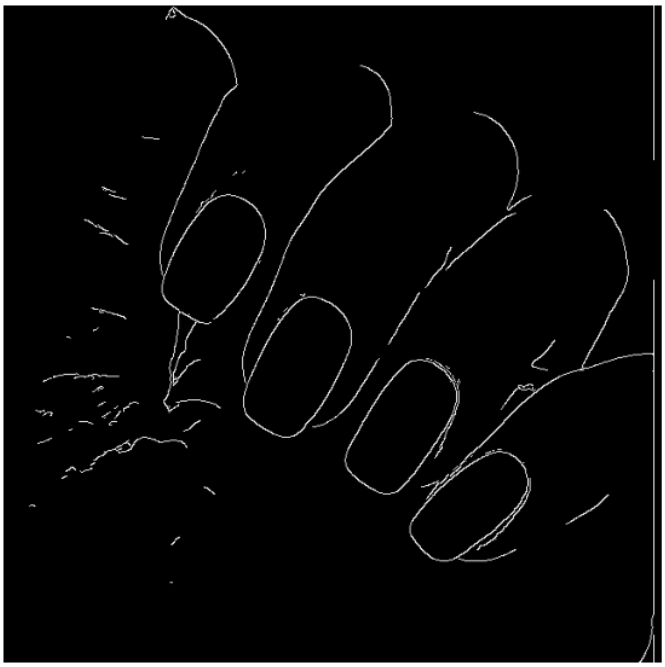
III. Implementasi Solusi

Program dibuat dengan menggunakan bahasa pemrograman Python, Pustaka OpenCV, NumPy, dan Matplotlib. Masukkan program adalah (1) satu buah citra jari pengguna dengan kuku yang terlihat jelas serta (2) satu buah citra warna/pola cat kuku. Keluaran program adalah satu buah citra berupa citra masukkan jari pengguna yang telah diberi cat kuku.

Gambar 12 Citra masukkan pertama: citra jari pengguna dengan kuku yang terlihat jelas

Pertama-tama program akan mengubah citra masukkan pertama dari citra RGB menjadi citra *grayscale*. Kemudian dilakukan pelembutan pada citra *grayscale* ini dan dilakukan pendeteksian tepi dengan operator Laplacian. Tujuan dilakukan pelembutan adalah supaya tidak banyak tepi yang terdeteksi. Dari proses ini didapatkan citra tepian berupa *citra grayscale* dimana beberapa detil dari jari (kerutan, garis pada latar belakang) sudah tidak terlalu terlihat. Pendeteksian tepi dengan operator Canny kemudian dilakukan pada citra tepian hasil operator Laplacian untuk mendapatkan citra biner dengan tepian satu garis lurus. Selanjutnya dilakukan pelembutan citra lagi dengan tujuan untuk melebarkan garis-garis hasil deteksi tepi.

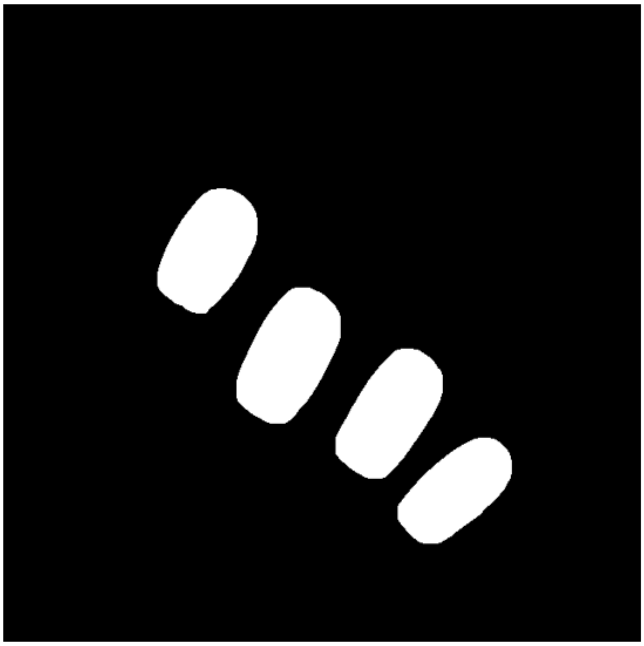
Gambar 13 Citra hasil pelembutan dan deteksi tepi menggunakan operator Laplacian



Gambar 14 Citra hasil deteksi tepi dengan operator Canny

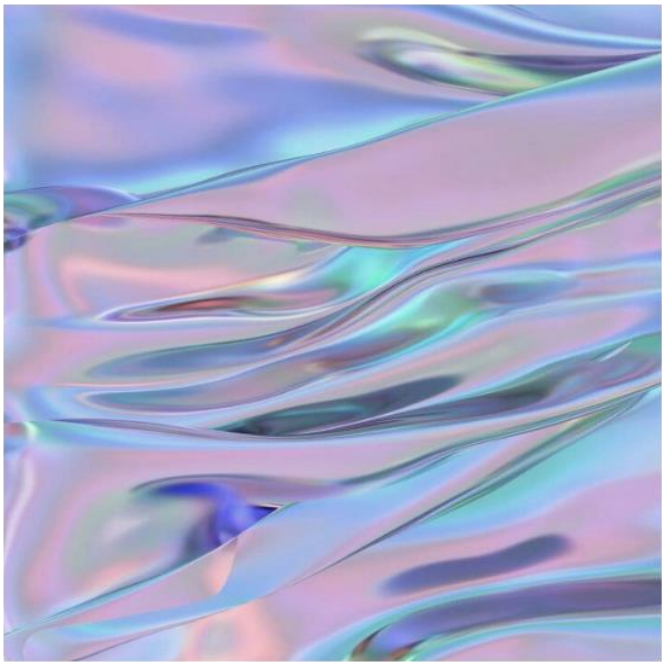
Selanjutnya dilakukan pembuatan kontur dari hasil deteksi tepi dan *pixel-pixel* pada bagian dalam kontur diisi dengan *pixel* putih. Setelah proses ini selesai, bagian kuku jari sudah tersegmentasi dengan baik, namun masih terdapat daerah-daerah gangguan berupa *pixel-pixel* putih berukuran kecil. Maka dari itu, digunakan fungsi *bwareaopen* dari Matlab yang dikonversi menjadi bahasa pemrogramman Python. Selain itu juga dilakukan pelebaran area kuku dengan fungsi *cv2.dilate* supaya lebih menutup gambar kuku di jari pengguna nantinya. Hasil dari rangkaian proses ini adalah sebuah *mask* citra biner pada Gambar 16.

Gambar 15 Citra hasil pelebaran dengan pelembutan citra

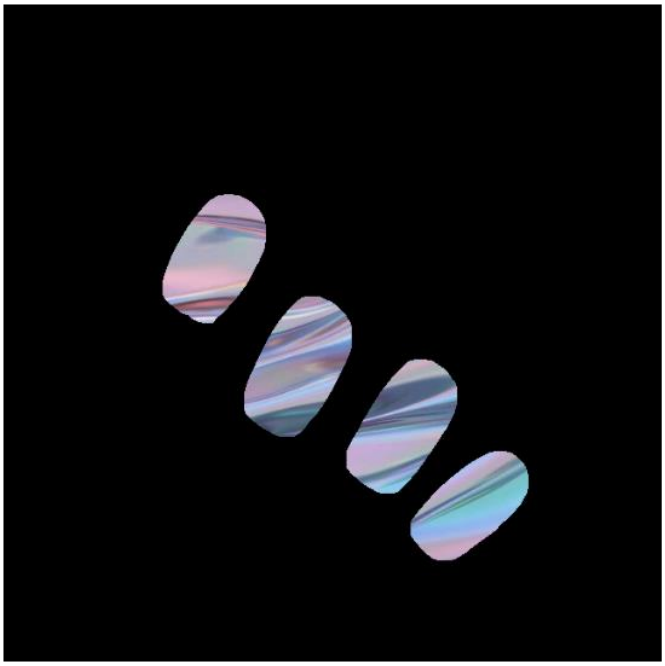


Gambar 16 Citra biner sebagai mask yang mengidentifikasi lokasi kuku pada jari pengguna

Selanjutnya *mask* yang dihasilkan akan digunakan untuk memotong citra masukkan kedua (citra warna/pola cat kuku) sehingga membentuk bentuk kuku pengguna.



Gambar 17 Citra masukkan kedua: warna/pola cat kuku



Gambar 18 Citra warna/pola cat kuku yang telah dipotong menggunakan mask

Tahap sealnjutnya adalah menggabungkan citra warna/pola kuku yang sudah terpotong ke citra masukkan jari dari tangan pengguna. Dengan menggunakan *mask*, dilakukan operasi penjumlahan pada daerah yang telah diidentifikasi sebagai kuku

untuk mengeliminasi *pixel* hitam. Hasil dari proses penjumlahan ini adalah Gambar 19, namun gambar ini tidak terlihat natural.

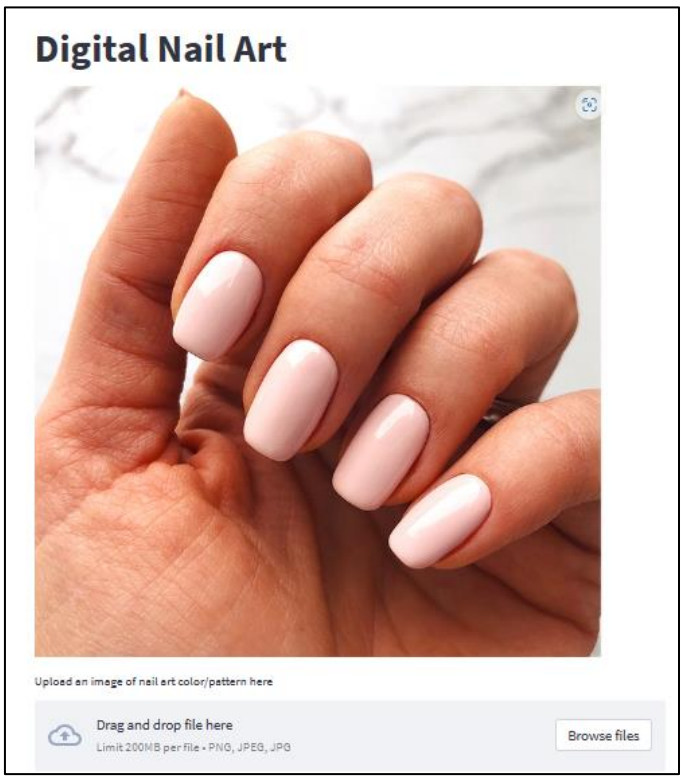
Gambar 19 Hasil penjumlahan citra kuku dengan citra warna/pola cat kuku dengan bantuan mask

Oleh karena itu, dilakukan pentransparansian *pixel* pada masing-masing citra masukkan sebelum akhirnya dijumlahkan menjadi satu citra keluaran menggunakan fungsi *cv2.addWeighted*. Hasil dari proses ini adalah citra pada Gambar 20 dan citra ini akan digunakan sebagai keluaran program.

Gambar 20 Citra keluaran

IV. Uji Coba Program

Tampilan antar muka menggunakan Streamlit dibuat untuk program ini dengan tampilan seperti pada Gambar 21. Pada tampilan antar muka ini user perlu memasukkan citra warna/pola cat kuku agar program bisa mengeluarkan citra keluaran berupa kuku yang telah diberi cat kuku.



Gambar 21 Tampilan antar muka program

Gambar 22 Program diuji dengan beberapa macam citra warna/pola cat kuku

Untuk memvalidasi performa program, program diuji dengan berbagai citra warna/pola cat kuku. Selain berbeda warna maupun pola, citra uji warna/pola juga memiliki ukuran resolusi yang berbeda-beda. Dari 10 variasi citra warna/pola cat kuku, program menghasilkan hasil yang memuaskan.

V. PENUTUP

A. *Kesimpulan*

Program untuk melihat hasil aplikasi cat kuku dengan bahasa pemrogramman Python, serta Pustaka OpenCV, NumPy dan Matplotlib telah berhasil dibuat. Hasil dari pengujian menunjukkan program dapat dengan baik mengaplikasikan warna/pola cat kuku ke jari pengguna

B. *Saran*

Meskipun program sudah dengan baik mengaplikasikan warna/pola cat kuku ke jari pengguna, beberapa variasi tambahan dapat ditambahkan kepada program seperti mengambil citra tangan secara langsung serta menambahkan variasi bentuk dan aksesoris pada *nail art* untuk lebih mengakomodasi kebutuhan penyedia jasa *nail art.* Selain itu program juga bisa dikembangkan untuk memberikan warna yang berbeda pada setiap kuku.

C. *Ucapan Terima Kasih*

Penulis berterima kasih terutama kepada Bapak Rinaldi Munir selaku dosen pengampu mata kuliah IF4073 Interpretasi dan Pengolahan Citra atas kontribusi beliau mulai dari membantu penulis yang merupakan mahasiswa non-informatika dalam pendaftaran mata kuliah ini, memberikan bahan ajar yang sangat mudah dipahami, hingga dukungan beliau bagi penulis untuk menyelesaikan mata kuliah ini. Penulis secara pribadi banyak belajar dari beliau baik untuk menyelesaikan mata kuliah ini maupun untuk keperluan tugas akhir penulis. Penulis juga mengucapkan terima kasih kepada asisten serta rekan-rekan yang bekerja selama penulis selama pelaksanaan mata kuliah ini.

REFERENSI

[1] Munir, Rinaldi. 2022. Pembentukan Citra dan Digitalisasi Citra. https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/03-Pembentukan-Citra-dan-Digitalisasi-Citra-2022.pdf

[2] Munir, Rinaldi. 2022. Pendeteksian Tepi Bagian 1. https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/18-Pendeteksian-Tepi-Bagian1-2022.pdf

[3] Munir, Rinaldi. 2022. Pendeteksian Tepi Bagian 2. https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/19-Pendeteksian-Tepi-Bagian2-2022.pdf

[4] Munir, Rinaldi. 2022. Image Enhancement Bagian 3. https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2022-2023/10-Image-Enhancement-Bagian3-2022.pdf

[5] Munir, Rinaldi. 2022. Citra Biner. https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/20-Citra-Biner-2021.pdf

[6] Munir, Rinaldi. 2022. Kontur. https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/21-Kontur-2021.pdf

[7] Munir, Rinaldi. 2022. Segmentasi Citra (Bagian 1). https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/22-Segmentasi-Citra-Bagian1-2022.pdf

[8] Munir, Rinaldi. 2022. Segmentasi Citra (Bagian 2). https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/22-Segmentasi-Citra-Bagian1-2022.pdf

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, 19 Desember 2022

Mellisa Irawan
13619063