

LAPORAN TUGAS

PENGOLAHAN CITRA DIGITAL

Pemrosesan Citra Warna dan Pengubahan Warna Wajah Menggunakan C++ dan OpenCV

(Gonzalez & Woods, Bab 6: Color Image Processing)

Nama : Efraim Melvyn Rafelino Lahai
NRP : 5803024009
Program Studi : Informatika
Mata Kuliah : Pengolahan Citra Digital
Dosen Pengajar : Ir. Drs. Peter Rhatodirdjo Angka, M.Kom., IPM., ASEAN Eng.

20 September 2026

Daftar Isi

1	Pendahuluan	2
1.1	Latar Belakang	2
1.2	Tujuan	2
1.3	Batasan	2
2	Landasan Teori	2
2.1	Komplemen Warna	2
2.2	Penyesuaian Nada dengan Kurva S	2
2.3	Ekualisasi Histogram pada Kanal Intensitas	3
2.4	Pengirisan Warna (<i>Color Slicing</i>) Berbentuk Bola	3
2.5	Penajaman Citra Warna dengan Laplacian	3
2.6	Pemrosesan Pseudowarna	3
2.7	Deteksi Wajah dengan Haar Cascade	3
2.8	Segmentasi Kulit di Ruang YCrCb	4
3	Perancangan dan Implementasi	4
3.1	Lingkungan Pengembangan	4
3.2	Alur Program	4
3.3	Fungsi Utama	4
3.4	Pengubahan Warna Wajah	5
3.5	Peningkatan Ketahanan Deteksi terhadap Cahaya	5
4	Hasil dan Pembahasan	5
4.1	Hasil Operasi Bab 6	6
4.2	Hasil Pengubahan Warna Wajah	8
4.3	Pengujian Kamera	9
4.4	Keterbatasan	9
4.5	Saran Pengembangan	9
5	Kesimpulan	9
A	Cara Kompilasi dan Menjalankan	10

1 Pendahuluan

1.1 Latar Belakang

Citra berwarna membawa informasi yang jauh lebih kaya dibandingkan citra keabuan. Bab 6 buku *Digital Image Processing* (Gonzalez & Woods) membahas model warna, transformasi warna, penghalusan dan penajaman citra berwarna, serta segmentasi berbasis warna. Tugas ini menerapkan konsep-konsep tersebut dalam sebuah program C++ yang memproses citra wajah, baik dari berkas gambar maupun dari kamera (*webcam*) secara langsung.

1.2 Tujuan

1. Mengimplementasikan beberapa operasi pemrosesan citra warna dari Bab 6.
2. Mendeteksi wajah pada citra atau video, lalu mengubah warna kulit wajah secara otomatis.
3. Menganalisis hasil dan keterbatasan program.

1.3 Batasan

Program menggunakan pustaka OpenCV untuk pembacaan citra, konversi ruang warna, dan deteksi wajah. Beberapa operasi dasar (misalnya pengirisan warna) ditulis manual per piksel, sedangkan operasi lain memakai fungsi bawaan OpenCV.

2 Landasan Teori

2.1 Komplemen Warna

Komplemen warna pada citra 8-bit ($L = 256$) dihitung per komponen:

$$s_i = (L - 1) - r_i, \quad i = B, G, R. \quad (1)$$

Operasi ini setara dengan negatif citra dan diterapkan pada setiap kanal.

2.2 Penyesuaian Nada dengan Kurva S

Kurva S meningkatkan kontras dengan menggelapkan nada gelap dan menerangkan nada terang. Program memakai fungsi sigmoid yang dinormalisasi agar $f(0) = 0$ dan $f(255) = 255$:

$$\sigma(r) = \frac{1}{1 + e^{-k(r-128)}}, \quad f(r) = 255 \frac{\sigma(r) - \sigma(0)}{\sigma(255) - \sigma(0)}, \quad k = 0,035. \quad (2)$$

Fungsi ini dihitung sekali dalam bentuk tabel pencarian (*look-up table*) berisi 256 nilai, lalu diterapkan ke seluruh piksel.

2.3 Ekualisasi Histogram pada Kanal Intensitas

Ekualisasi histogram langsung pada kanal R, G, dan B akan menggeser rona sehingga muncul warna palsu. Cara yang lebih aman adalah mengubah citra ke ruang warna yang memisahkan informasi warna dari kecerahan, lalu meratakan histogram hanya pada komponen kecerahan. Program memakai ruang HSV dan meratakan kanal *Value*, sebagai pendekatan terhadap kanal intensitas pada model HSI. Kanal *Saturation* dinaikkan 15%.

2.4 Pengirisan Warna (*Color Slicing*) Berbentuk Bola

Pengirisan warna mengisolasi warna tertentu. Untuk daerah berbentuk bola dengan pusat $\mathbf{a} = (a_B, a_G, a_R)$ dan jari-jari R_0 , piksel $\mathbf{z}$ dipertahankan jika jaraknya ke pusat tidak melebihi R_0 , dan diganti dengan warna netral jika sebaliknya:

$$s_i = \begin{cases} 128 & \text{jika } \sum_j (z_j - a_j)^2 > R_0^2, \\ r_i & \text{lainnya.} \end{cases} \quad (3)$$

Nilai 128 adalah padanan 8-bit dari abu-abu netral 0,5.

2.5 Penajaman Citra Warna dengan Laplacian

Laplacian vektor pada citra warna sama dengan Laplacian yang dihitung terpisah pada tiap komponen warna. Penajaman dilakukan dengan mengurangkan Laplacian dari citra asli, $g = f - \nabla^2 f$. Untuk Laplacian 4-tetangga, hal ini menghasilkan kernel

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (4)$$

2.6 Pemrosesan Pseudowarna

Pseudowarna menugaskan warna pada tiap tingkat keabuan sehingga perbedaan intensitas yang halus lebih mudah dilihat. Program mengubah citra ke skala keabuan lalu menerapkan peta warna JET.

2.7 Deteksi Wajah dengan Haar Cascade

Deteksi wajah memakai *Haar cascade classifier* (metode Viola-Jones) yang tersedia di OpenCV, berkas `haarcascade_frontalface_default.xml`. Pengklasifikasi ini memindai citra keabuan pada beberapa skala dan mengembalikan kotak pembatas untuk wajah menghadap depan.

2.8 Segmentasi Kulit di Ruang YCrCb

Warna kulit manusia mengelompok pada rentang sempit di komponen krominansi C_r dan C_b dan relatif tidak bergantung pada kecerahan (Y). Karena itu piksel kulit dapat dipisahkan dengan ambang sederhana pada C_r dan C_b .

3 Perancangan dan Implementasi

3.1 Lingkungan Pengembangan

- Bahasa: C++17 (kompilator MinGW-w64 GCC 16.1, UCRT).
- Pustaka: OpenCV 4.10.0, dikompilasi dari kode sumber dengan modul `core`, `imgproc`, `imgcodecs`, `videoio`, `highgui`, dan `objdetect`.
- Sistem *build*: CMake dan Ninja.
- Sistem operasi: Windows 11.

3.2 Alur Program

1. Program dijalankan dengan jalur berkas gambar, atau dengan `-cam` (atau tanpa argumen) untuk mode kamera.
2. Berkas *Haar cascade* dimuat.
3. **Mode berkas:** enam operasi Bab 6 dijalankan pada gambar masukan, lalu wajah dideteksi dan warnanya diubah. Semua hasil disimpan dalam format PNG ke folder `output_results`.
4. **Mode kamera:** setiap bingkai dideteksi wajahnya dan diubah warnanya, lalu ditampilkan. Tombol `s` menyimpan bingkai, tombol `q` atau `Esc` keluar.

3.3 Fungsi Utama

Tabel 1: Fungsi pada namespace `ColorProcessing`.

Fungsi	Keterangan
<code>computeComplement</code>	Komplemen warna dengan <code>bitwise_not</code> .
<code>adjustToneSCurve</code>	Kurva S berbasis tabel pencarian.
<code>equalizeIntensityHSI</code>	Ekualisasi kanal <i>Value</i> di HSV.
<code>sliceColorSphere</code>	Pengirisan warna bola 3D, ditulis manual per piksel.
<code>sharpenLaplacian</code>	Penajaman dengan kernel Laplacian 3×3 .
<code>generatePseudocolor</code>	Keabuan lalu peta warna JET.
<code>detectFaces</code>	Deteksi wajah dengan Haar cascade.
<code>recolorFace</code>	Pengubahan warna kulit di dalam wajah.

3.4 Pengubahan Warna Wajah

Fungsi `recolorFace` bekerja pada setiap kotak wajah hasil deteksi:

1. Bagian wajah diubah ke ruang YCrCb, lalu piksel kulit dipilih dengan ambang $C_r \in [135, 195]$ dan $C_b \in [80, 135]$.
2. Hasilnya dikalikan (operasi AND) dengan elips yang mengisi kotak wajah, sehingga latar di sudut kotak tidak ikut berubah.
3. Masker dihaluskan dengan *Gaussian blur* 15×15 agar tepi transisinya lembut.
4. Bagian wajah diubah ke HSV. Kanal *Hue* diputar sebesar 60 (skala OpenCV 0–179, sehingga kulit oranye menjadi hijau) dan *Saturation* dikalikan 1,3.
5. Hasil putaran warna dicampur dengan citra asli menurut masker: $\text{keluaran} = \alpha \cdot \text{berubah} + (1 - \alpha) \cdot \text{asli}$.

Karena hanya piksel kulit di dalam wajah yang bobotnya besar, rambut, mata, dan latar tetap berwarna asli.

3.5 Peningkatan Ketahanan Deteksi terhadap Cahaya

Pengujian awal menunjukkan deteksi wajah gagal pada citra dengan pencahayaan kurang merata. Ekualisasi histogram global (`equalizeHist`) diganti dengan CLAHE (*Contrast Limited Adaptive Histogram Equalization*, batas kontras 3,0, petak 8×8) sebelum deteksi. Setelah perubahan ini, foto uji yang sebelumnya menghasilkan 0 wajah kini menghasilkan 1 wajah.

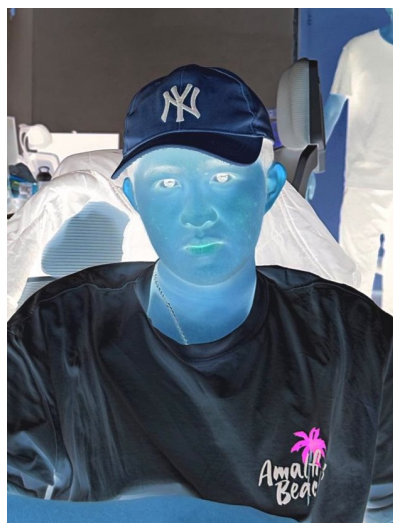
4 Hasil dan Pembahasan

Pengujian memakai foto penulis (Gambar 1) dan citra uji standar *Lena* yang disertakan dalam kode sumber OpenCV.



Gambar 1: Citra masukan (foto penulis).

4.1 Hasil Operasi Bab 6



(a) Komplemen warna

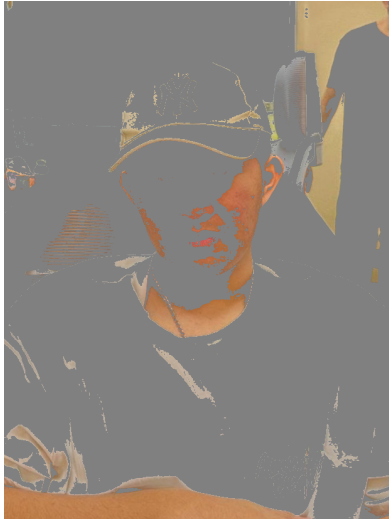


(b) Kurva S



(c) Ekualisasi kanal V

Gambar 2: Transformasi warna dan nada.



(a) Pengirisan warna



(b) Penajaman Laplacian



(c) Pseudowarna JET

Gambar 3: Pengirisan warna, penajaman, dan pseudowarna.

Komplemen membalik seluruh warna: kulit menjadi kebiruan dan putih menjadi gelap. **Kurva S** menambah kontras nada tengah. **Ekualisasi kanal Value** meratakan sebaran kecerahan tanpa menggeser rona. **Penajaman Laplacian** mempertegas tepi dan tekstur. **Pseudowarna** mengubah tingkat keabuan menjadi warna sehingga daerah terang dan gelap lebih mudah dibedakan.

Pada **pengirisan warna**, hanya sebagian kulit wajah yang dipertahankan, sedangkan sisanya menjadi abu-abu. Hal ini terjadi karena pusat bola ($BGR = 110, 140, 190$) dan jari-jari 65 bersifat tetap, sehingga bagian wajah yang lebih terang atau lebih gelap karena bayangan jatuh di luar bola. Beberapa objek berwarna hangat di latar (misalnya dinding krem) juga ikut lolos. Ini menunjukkan bahwa pengirisan warna dengan prototipe tetap peka terhadap pencahayaan dan warna kulit.

4.2 Hasil Pengubahan Warna Wajah



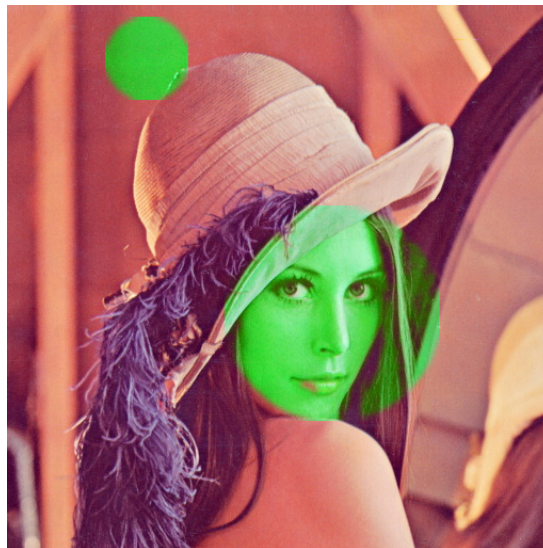
(a) Sebelum



(b) Sesudah

Gambar 4: Pengubahan warna wajah pada foto penulis (1 wajah terdeteksi).

Wajah terdeteksi dan kulit berubah menjadi hijau, sedangkan baju, kursi, dan latar tidak berubah. Pada tepi tepi terlihat sedikit rona hijau karena warna topi krem masuk ke dalam rentang kulit dan berada di dalam kotak wajah.



Gambar 5: Pengubahan warna wajah pada citra uji *Lena*.

Pada citra *Lena*, wajah utama berubah warna dengan baik. Dengan parameter deteksi yang sekarang, program mengembalikan 2 kotak wajah pada citra ini, sehingga ada satu deteksi tambahan (positif palsu).

4.3 Pengujian Kamera

Mode kamera bekerja dan menampilkan wajah dengan warna yang sudah diubah secara langsung. Namun pada ruangan yang kurang terang, wajah awalnya tidak terdeteksi; deteksi baru berhasil setelah wajah diberi cahaya tambahan dari senter ponsel. Pengujian kamera setelah penggantian ke CLAHE belum dilakukan pada kondisi redup yang sama.

4.4 Keterbatasan

1. **Bergantung pada pencahayaan.** Haar cascade dilatih pada wajah menghadap depan dengan pencahayaan cukup. Cahaya redup, bayangan kuat, wajah menunduk atau miring, serta penutup seperti topi dapat menggagalkan deteksi.
2. **Positif palsu.** Menurunkan ambang deteksi meningkatkan kemungkinan objek non-wajah terdeteksi sebagai wajah.
3. **Ambang kulit tetap.** Rentang C_r/C_b yang tetap tidak cocok untuk semua warna kulit dan pencahayaan, dan dapat mengenai objek berwarna serupa (topi, kayu, dinding).
4. **Pengirisan warna** memakai prototipe tetap, sehingga hasilnya hanya baik untuk satu nada kulit.
5. **Bukan pelacakan.** Setiap bingkai dideteksi ulang tanpa pelacakan, sehingga kotak dapat berkedip antarbingkai.

4.5 Saran Pengembangan

Detektor wajah berbasis jaringan saraf (misalnya model DNN pada OpenCV) lebih tahan terhadap cahaya dan sudut pandang dibandingkan Haar cascade. Ambang kulit juga dapat dibuat adaptif, misalnya dengan mengambil statistik warna dari bagian tengah wajah yang terdeteksi.

5 Kesimpulan

Program berhasil menerapkan enam operasi pemrosesan citra warna dari Bab 6 (komplemen, kurva S, ekualisasi kanal intensitas, pengirisan warna, penajaman Laplacian, dan pseudowarna) serta mendeteksi wajah dan mengubah warna kulitnya pada gambar dan pada kamera. Pengubahan warna paling baik pada wajah menghadap depan dengan pencahayaan cukup. Penggantian ekualisasi global dengan CLAHE meningkatkan deteksi pada foto uji, tetapi deteksi pada cahaya redup dan positif palsu masih menjadi keterbatasan.

A Cara Kompilasi dan Menjalankan

```
cmake -S . -B build -G Ninja -DCMAKE_BUILD_TYPE=Release ^  
      -DOpenCV_DIR=D:/opencv/install/x64/mingw/lib  
cmake --build build  
  
cd build  
.\color_processor.exe path\ke\foto.jpg      # mode berkas  
.\color_processor.exe --cam                 # mode kamera
```