

Aplikasi Foto Mozaik Adaptif

Dion Saputra - 13516045

Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
E-mail: dionsaputra666@gmail.com

Abstract. Foto mozaik merupakan salah satu karya seni dalam bidang fotografi yang diperoleh dengan menyusun foto-foto kecil sehingga membentuk suatu foto yang lebih besar yang dapat diinterpretasi oleh manusia. Secara konvensional foto mozaik dibuat secara manual oleh manusia. Hal ini tentu saja menyulitkan baik dari segi waktu maupun tenaga. Makalah ini membahas mengenai pembuatan foto mozaik secara otomatis. Metode yang digunakan dalam penyelesaian masalah tersebut adalah menggunakan pengubinan adaptif, pencocokan ubin, dan koreksi warna. Hasil yang diperoleh dapat digunakan dengan baik sesuai dengan parameter yang digunakan.

1. Pendahuluan

Foto mozaik merupakan foto yang dibentuk sedemikian rupa menggunakan foto-foto lain yang lebih kecil sehingga menghasilkan bentuk baru yang dapat diinterpretasi. Efek yang dihasilkan oleh foto mozaik didapatkan dari kombinasi warna-warna yang ada pada foto-foto kecil penyusunnya. Foto mozaik dapat dibuat secara manual dengan mencocokkan blok warna pada foto utama dengan kandidat foto penyusunnya. Namun, metode manual tentunya membutuhkan waktu dan usaha yang besar untuk dapat menghasilkan sebuah foto mozaik.

Beberapa metode telah dikembangkan untuk menghasilkan foto mozaik secara otomatis. Metode secara sederhana yang dapat digunakan adalah mengambil foto penyusun dengan warna terdekat pada blok-blok tertentu yang berukuran tetap. Lee (2017), mengusulkan algoritma foto mozaik dengan *adaptive tiling* dan koreksi nilai intensitas. Usulan algoritma tersebut mendapatkan hasil yang lebih baik dibandingkan dengan algoritma sebelumnya.

Pada makalah ini, penulis mengusulkan untuk melakukan percobaan foto mozaik dengan pengurutan berdasarkan nilai *hue* dan koreksi warna menggunakan pencampuran citra pada algoritma foto mozaik yang telah dikembangkan Lee. Dengan adanya pengurutan nilai *hue* dan koreksi warna tersebut pada foto mozaik, diharapkan dapat menghasilkan foto mozaik yang lebih baik.

2. Dasar Teori

2.1. Citra

Citra adalah sinyal dwimatra yang bersifat menerus (*continue*) yang dapat ditangkap oleh sistem visual manusia [1]. Citra digital merupakan representasi dari citra yang dapat disimpan dan diproses menggunakan komputer. Bentuk representasi visual lain yang sering dikaitkan dengan citra digital yaitu grafik (*graphic*). Hal utama yang membedakan citra digital dengan grafik yaitu citra ditangkap

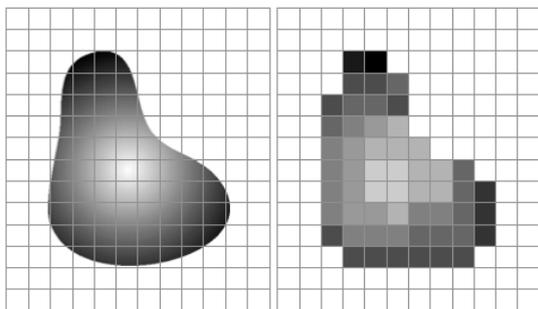
langsung dari sinyal cahaya lalu ditransformasi ke dalam bentuk digital. Pada grafik representasi digital yang ada diperoleh dari aplikasi atau program pembuat grafik.

Sinyal cahaya tampak merupakan fungsi dwimatra dari intensitas cahaya [2]. Ketika sinyal cahaya tampak tersebut mengenai suatu permukaan benda, sinyal tersebut akan dipantulkan dengan pengaruh dari permukaan benda yang dikenai. Ketika sinyal pantulan diterima oleh mata manusia, persepsi visual dari benda tersebut baru dapat dilihat. Keterkaitan antara fungsi intensitas cahaya $i(x,y)$, fungsi refleksi/pantulan permukaan benda $r(x,y)$, dan sinyal pantulan yang diterima mata manusia dapat direpresentasikan dalam persamaan (1) [2]. Jangkauan nilai dari fungsi $i(x,y)$ adalah $[0, \infty)$ dan jangkauan nilai dari fungsi $r(x,y)$ adalah $[0,1]$ dengan nilai 1 menyatakan pemantulan sempurna [2].

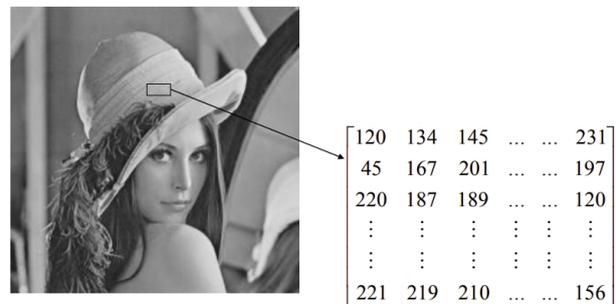
$$f(x, y) = i(x, y) \cdot r(x, y) \quad (1)$$

Penangkapan citra dari sinyal cahaya tampak yang kontinu dapat dilakukan menggunakan perangkat elektronik tertentu. Perangkat elektronik yang digunakan untuk menangkap sinyal citra dapat beragam, tergantung pada keperluan. Untuk keperluan penangkapan citra sehari-hari, citra dapat ditangkap menggunakan kamera. Sedangkan pada keperluan yang lebih spesifik, misalnya pada dunia medis, penangkapan citra dapat dilakukan dengan mesin *rontgen*.

Sinyal citra yang menerus tidak efisien untuk disimpan dan diproses pada komputer sehingga dibutuhkan transformasi ke bentuk diskrit. Transformasi yang digunakan untuk mengubah sinyal cahaya tampak yang menerus ke dalam citra digital menggunakan metode *sampling* dan kuantisasi [2]. *Sampling* spasial pada sinyal cahaya tampak yang menerus merubah representasi citra digital menjadi unit-unit intensitas cahaya yang disebut dengan *pixel*. Semakin banyak jumlah *pixel* yang dihasilkan dari proses *sampling* yang dilakukan, citra digital yang diperoleh akan semakin mendekati bentuk visual citra yang sebenarnya. Gambar 1 menunjukkan ilustrasi *sampling* dari sinyal citra menerus ke dalam kumpulan *pixel* pada citra digital. Proses kuantisasi berkaitan dengan transformasi nilai intensitas sinyal ke dalam bilangan bulat taknegatif agar dapat disimpan dalam *bit* memori komputer.



Gambar 1: Proses Sampling pada Citra Digital



Gambar 2: Citra Digital Sebagai Matriks [1]

Salah satu cara merepresentasikan citra digital pada komputer adalah menggunakan struktur data matriks. Setiap elemen matriks yang digunakan berkorespondensi dengan nilai *pixel* pada koordinat (i,j) pada citra digital. Setiap elemen matriks memiliki jangkauan nilai tertentu sesuai hasil kuantisasi. Sebagai contoh penggunaan 8 *bit* memori untuk menyimpan nilai *pixel* menghasilkan jangkauan elemen matriks dari 0 hingga 255. Gambar 2 menunjukkan ilustrasi representasi matriks dari sebuah citra digital.

2.2. Warna

Gelombang cahaya tampak merupakan rentang gelombang elektromagnetik yang dapat dipersepsi oleh sistem visual manusia. Gelombang ini berada pada panjang gelombang sekitar 400 hingga 700 nm dengan setiap rentang panjang gelombang memiliki karakteristik warna tertentu [3]. Tabel 1 menunjukkan karakteristik warna pada masing-masing rentang gelombang di spektrum cahaya

tampak. Saat cahaya tampak mengenai permukaan benda, tidak semua gelombang warna akan dipantulkan kembali oleh permukaan tersebut. Permukaan dengan warna tertentu akan dominan memantulkan gelombang warna yang berkorespondensi. Saat mata manusia menerima gelombang cahaya pantulan tersebut, manusia dapat mengenali warna dari permukaan yang dilihat.

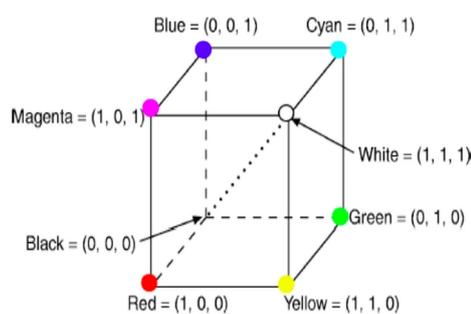
Tabel 1: Karakteristik Panjang Gelombang Warna [3]

| Warna | Panjang Gelombang (nm) |
|--------|------------------------|
| Ungu | 390 - 430 |
| Nila | 440 - 450 |
| Biru | 460 - 480 |
| Hijau | 490 - 540 |
| Kuning | 550 - 580 |
| Jingga | 590 - 640 |
| Merah | 650 - 800 |

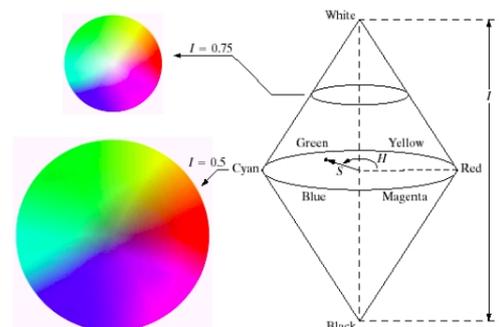
Warna dalam citra digital direpresentasikan melalui model warna. Terdapat banyak model warna yang telah didefinisikan hingga saat ini. Pada tulisan ini hanya akan dibahas dua diantaranya yang dipakai pada penyelesaian masalah, yaitu model warna RGB dan HSI

Model warna RGB menggunakan tiga komponen warna dasar merah, hijau, dan biru yang berturut-turut disimbolkan dengan R, G, dan B. Setiap komponen memiliki rentang nilai [0,1]. Warna-warna lain diperoleh melalui kombinasi linear dari ketiga komponen warna dasar tersebut seperti pada persamaan 2. Model warna RGB dapat direpresentasikan melalui kubus warna seperti pada Gambar 3.

$$C = aR + bG + cB \quad (2)$$



Gambar 3: Kubus Warna RGB [1]



Gambar 4: Kerucut Warna HSI [1]

Model warna HSI merepresentasikan warna dalam citra digital seperti halnya manusia melakukan persepsi terhadap warna. Model warna HSI terdiri dari tiga komponen warna yaitu *hue*, *saturation*, dan *intensity*. Komponen *hue* menyatakan nilai sebenarnya dari warna tersebut, yang berkorespondensi dengan panjang gelombang dari warna tersebut [1]. Komponen *saturation* menyatakan tingkat kemurnian warna cahaya yang mengindikasikan seberapa banyak warna putih yang diberikan [1]. Komponen *intensity* menyatakan banyaknya cahaya yang diterima oleh mata tanpa mepedulikan warna [1]. Model HSI dapat direpresentasikan melalui kerucut warna HSI seperti pada Gambar 4.

Model warna HSI dapat diperoleh dari transformasi model warna RGB. Transformasi model warna RGB ke model warna HSI dapat diperoleh melalui persamaan 3, 4, dan 5 [1].

$$H = \arccos\left(\frac{2R - G - B}{2\sqrt{(R - G)^2(R - G)(R - B)}}\right) \quad (3)$$

$$S = 1 - \frac{3}{R + G + B} \min(R, G, B) \quad (4)$$

$$I = (R + G + B) / 3 \quad (5)$$

Sebaliknya, transformasi balikan dari HSI ke RGB dapat diperoleh melalui transformasi pada persamaan 6 [1]

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \sqrt{3}/3 & 0 & 2/\sqrt{6} \\ \sqrt{3}/3 & 1/\sqrt{2} & -1/\sqrt{6} \\ \sqrt{3}/3 & -1/\sqrt{2} & -1/\sqrt{6} \end{bmatrix} \begin{bmatrix} I \\ S \cos(H) \\ S \sin(H) \end{bmatrix} \quad (6)$$

2.3. Penskalaan Citra

Permasalahan penskalaan citra terbagi atas dua submasalah, yaitu perbesaran citra dan pengecilan citra. Pada permasalahan pengecilan citra, total elemen dalam matriks yang disimpan citra menjadi berkurang. Permasalahan ini dapat diselesaikan dengan membuang nilai *pixel* yang tidak berkorespondensi dengan koordinat hasil pengecilan. Submasalah pada perbesaran citra, nilai-nilai *pixel* baru perlu ditambahkan untuk mengisi *pixel* yang kosong setelah diperbesar. Terdapat tiga algoritma yang dapat digunakan, yaitu *nearest neighbor*, interpolasi bilinear, dan interpolasi bikubik [4]. Algoritma *nearest neighbor* mengisi nilai *pixel* yang kosong dengan nilai *pixel* tetangga yang sudah terisi. Algoritma interpolasi bilinear menggunakan interpolasi dari 4 *pixel* terdekat untuk menentukan nilai *pixel* yang kosong. Sedangkan algoritma interpolasi bikubik menggunakan interpolasi dari 16 *pixel* terdekat. Algoritma *nearest neighbor* merupakan algoritma paling cepat sedangkan algoritma interpolasi bikubik memberikan hasil yang paling mulus.



Gambar 5: Operasi Perbesaran Citra

2.4. Pencampuran Citra

Pencampuran citra merupakan operator untuk menggabungkan dua buah citra menjadi citra baru dimana karakteristik dari dua citra asal masih terlihat pada citra hasil. Operator pencampuran citra ini merupakan operator aras titik menurut persamaan 7 [5]. Konstanta a berada dalam rentang $[0,1]$ yang menyatakan seberapa besar kontribusi dari gambar f terhadap gambar hasil. Gambar 6 memperlihatkan satu buah citra (kanan) dihasilkan dari dua buah citra lainnya.

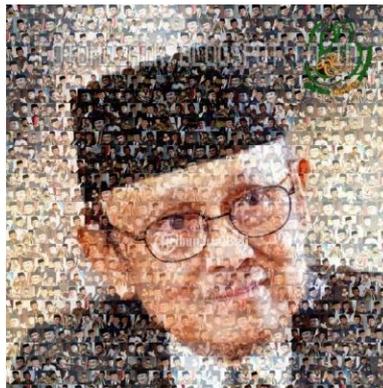
$$h(x, y) = a * f(x, y) + (1 - a) * g(x, y) \quad (7)$$



Gambar 6: Pencampuran Citra [5]

2.5. Foto Mozaik

Foto mozaik merupakan salah satu jenis karya seni dalam bidang fotografi dimana banyak gambar disusun sedemikian rupa sehingga menghasilkan suatu bentuk yang dapat dikenali. Foto mozaik terdiri atas satu buah citra acuan dan banyak citra penyusun. Gambar 7 memperlihatkan salah satu contoh foto mozaik.



Gambar 7: Contoh Foto Mozaik

3. Batasan Masalah

Pada makalah ini terdapat beberapa pembatasan masalah yang ingin diselesaikan. Batasan masalah tersebut yaitu sebagai berikut

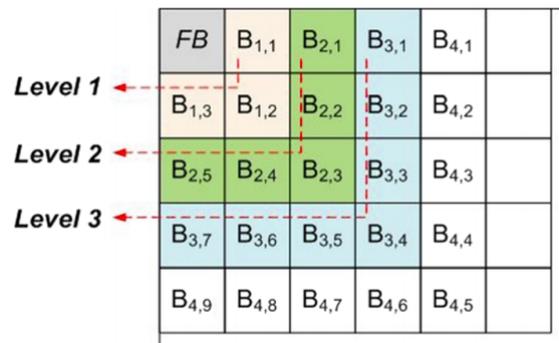
- Citra acuan dan citra penyusun sudah disimpan dalam bentuk berkas citra. Citra yang masih dalam *preview* kamera atau citra yang masih dalam *frame* video berada di luar lingkup penyelesaian masalah.
- Citra acuan dan citra penyusun memiliki panjang dan lebar yang sama. Hal ini dilakukan untuk mempermudah implementasi utama dari program foto mozaik yang dibuat.

4. Ide Penyelesaian Masalah

4.1. Pengubinan Adaptif

Pengubinan adaptif mengadaptasi algoritma yang dikembangkan pada [6] dengan mengganti indikator koreksi yang semula menggunakan perbedaan RGB menjadi menggunakan perbedaan komponen *hue*. Tujuan dari pengubinan adaptif ini adalah mengembangkan blok *pixel* sebesar mungkin sehingga nilai selisih *hue* dari masing-masing *pixel* tersebut tidak berbeda jauh.

Gambar 8 menunjukkan blok-blok *pixel* dari foto acuan pada keadaan awal. Blok *FB* merupakan *fixed block* yang merupakan titik referensi dari blok yang hasil ekspansi. Nilai *hue* pada *FB* akan dibandingkan dengan nilai *hue* pada blok-blok dalam suatu level. Apabila semua blok pada level *i* memiliki perbedaan nilai *hue* di bawah nilai ambang tertentu, blok *FB* akan diekspansi sehingga mencakup level *i*. Untuk selanjutnya, blok yang sudah diekspansi ini disebut dengan ubin. Proses tersebut diulangi hingga nilai maksimum level yang dibolehkan atau terdapat perbedaan *hue* di atas nilai ambang.



Gambar 8: Blok pada Penyusunan Adaptif

4.2. Pencocokan Ubin

Pencocokan ubin bertujuan untuk menemukan citra paling tepat untuk mengisi ubin-ubin hasil pengubinan adaptif. Citra yang tepat untuk mengisi ubin adalah citra yang memiliki selisih rerata *hue* paling kecil dengan nilai *hue* dari ubin. Namun hasil ini dapat membuat citra-citra yang sama terkumpul pada ubin-ubin yang berdekatan sehingga menyebabkan foto mozaik menjadi monoton. Untuk mengurangi efek ini, gambar diambil secara acak dari *k* gambar tercocok. Citra-citra yang sudah dipilih, diskalakan agar sesuai dengan ukuran ubin, lalu digambarkan pada ubin tersebut.

4.3. Koreksi Warna

Hasil pencocokan ubin sudah berupa foto mozaik. Namun untuk koleksi foto acuan yang sedikit, citra acuan menjadi tidak begitu jelas. Koreksi warna dapat dilakukan pada masing-masing komponen warna HSI atau menggunakan metode pencampuran citra. Pendekatan yang digunakan pada [6] untuk melakukan koreksi citra adalah menggunakan koreksi pada komponen warna intensitas. Sedangkan pendekatan yang digunakan pada makalah ini adalah pendekatan dengan pencampuran warna.

5. Implementasi Penyelesaian Masalah

5.1. Lingkungan Kerja

Lingkungan kerja dari implementasi penyelesaian masalah dan pengujian dapat dilihat pada tabel 2

Tabel 2: Lingkungan Kerja Implementasi dan Pengujian

| Kategori | Lingkungan Kerja |
|--------------------|--------------------------------|
| Sistem Operasi | Ubuntu 19.10 (Linux 64 bit) |
| Bahasa Pemrograman | Java 8u201 |
| Kakas IDE | Intellij IDEA Community 2019.3 |
| Ekstensi Citra Uji | .jpg |

5.2. Struktur Data

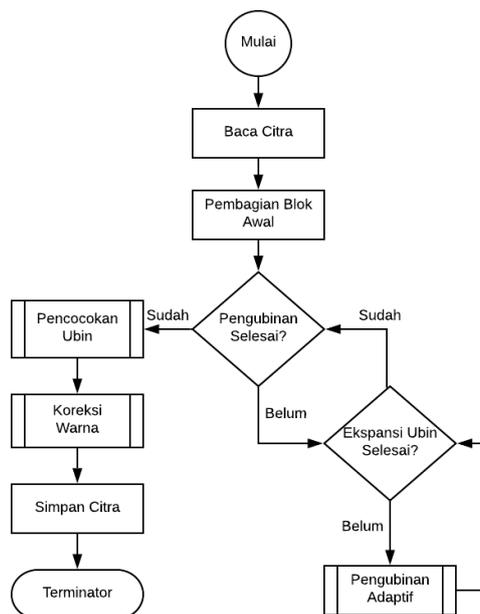
Struktur data pada tabel merupakan struktur data non-bawaan yang digunakan dalam tahapan implementasi.

Tabel 3: Struktur Data yang Digunakan

| Nama Kelas | Deskripsi | Fungsi atau prosedur penting |
|-------------|--|---|
| Pair | Pasangan data. Dapat berbeda tipe. | - |
| Block | Merepresentasikan persegi yang didefinisikan melalui titik kiri atas dan titik kanan bawahnya. | Menentukan apakah suatu blok merupakan <i>fixed block</i> atau tidak. |
| MozaicImage | Struktur data untuk menyimpan data citra. | Baca/tulis citra, konversi RGB-HSI, menghitung perbedaan <i>hue</i> blok dengan blok, dan menghitung perbedaan <i>hue</i> blok dengan citra penyusun. |

5.3. Program Utama

Alur program utama dari implementasi penyelesaian masalah dapat dilihat pada gambar 9



Gambar 9: Diagram Alir Implementasi

6. Pengujian

6.1. Ukuran Blok



Ukuran Blok 2 px



Ukuran Blok 8 px

6.2. Maksimum Level



Maksimum Level 2



Maksimum Level 8

6.3. Ambang Batas Hue



Ambang Batas Hue 0.2



Ambang Batas Hue 0.8

6.4. Koreksi Warna



Koreksi Warna 0.2



Koreksi Warna 0.8

7. Kesimpulan dan Saran

Terdapat empat parameter yang diuji yaitu ukuran blok awal, maksimum level, dan ambang batas *hue*, koreksi warna. Berdasarkan pengujian didapatkan beberapa kesimpulan berikut. Penggunaan parameter blok awal yang kecil memberikan hasil lebih akurat untuk merepresentasikan citra acuan karena dapat mengisi detail dari gambar. Penggunaan parameter maksimum level yang tinggi memberikan variasi pengubinan yang lebih beragam. Penggunaan ambang batas *hue* yang kecil lebih merepresentasikan citra acuan karena ambang batas *hue* yang kecil menunjukkan ubin dan citra acuan warnanya mirip. Penggunaan koreksi warna yang besar semakin memperjelas citra acuan, namun juga membuat citra penyusun menjadi lebih kabur. Secara keseluruhan, penggunaan parameter yang tepat dapat meningkatkan kualitas foto mozaik yang dihasilkan.

Adapun saran dari eksperimen ini adalah algoritma yang diimplementasikan masih belum begitu cepat. Beberapa optimasi dalam algoritma dibutuhkan untuk meningkatkan kecepatan pemrosesan.

8. Ucapan Terima Kasih

Pertama penulis mengucapkan syukur kepada Tuhan Yang Maha Esa, sehingga penulis dapat menyelesaikan penulisan makalah Interpretasi dan Pengolahan Citra Digital ini. Setelah itu penulis mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir sebagai dosen mata kuliah Interpretasi dan Pengolahan Citra Digital di program studi Teknik Informatika Institut Teknologi Bandung yang telah memberikan pengajaran terhadap mata kuliah Interpretasi dan Pengolahan Citra Digital di semester ini. Kemudian, penulis juga menyampaikan ucapan terima kasih kepada kedua orang tua penulis, keluarga, dan teman-teman yang telah membantu dari segi nonteknis. (2)

9. Referensi

- [1] Munir, Rinaldi. 2019. Bahan Ajar Kuliah IF4073 Interpretasi dan Pengolahan Citra Digital. Bandung: Institut Teknologi Bandung.
- [2] Gonzales, R.C., & Woods, R.E. (2002). Digital Image Processing.
- [3] <https://www.globalbeads.com/education/physics-of-color/> diakses pada 12 Desember 2019 pukul 16.40
- [4] Tabora, V. 2019. JPEG Image Scaling Algorithm. Url: <https://medium.com/hd-pro/jpeg-image-scaling-algorithms-913987c9d588> diakses pada 13 Desember 2019 10.00
- [5] OpenCV. 2019. Adding or Blending Two Images with OpenCV . Url: https://docs.opencv.org/2.4/doc/tutorials/core/adding_images/adding_images.html diakses pada 12 Desember 2019 13.00
- [6] Lee, H. Y. 2017. Automatic Photomosaic Algorithm Through Adaptive Tiling and Block Matching. *Multimedia Tools and Applications*.