

Penentuan Palet Warna pada Gambar Raster dengan Algoritma Divide and Conquer

Malvin Juanda / 13514044

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

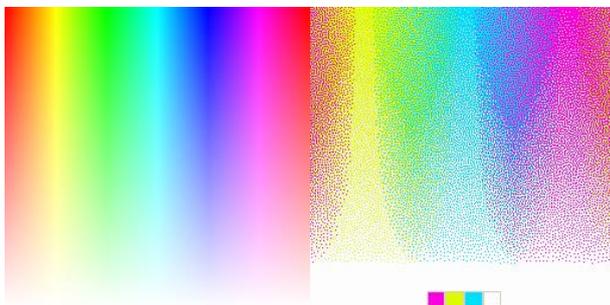
13514044@std.stei.itb.ac.id

Abstrak—*Color Pallete* merupakan komponen yang sangat penting dalam *digital design* misalnya dalam membuat brosur, banner, poster, dan sejenisnya. Penggunaan palet warna dalam *digital-editing* membuat penggunaan warna dalam suatu desain tidak terlalu berlebihan serta membangun koneksi antar objek pada desain. Penentuan *color pallete* biasanya dilakukan sebelum memulai desain dan warna yang dipilih bersesuaian dengan tema desain. Namun, kadangkala seorang *designer* dapat meneruskan proyek dari orang lain yang tidak mencantumkan palet warna di hasil seninya sehingga akan merepotkan *designer* tersebut. Oleh karena itu, dalam makalah ini akan dituliskan metode untuk menentukan palet warna pada suatu gambar raster dengan memanfaatkan algoritma *divide and conquer*.

Keywords—*raster, palet, color quantization, pixel*

I. PENDAHULUAN

Dalam pembuatan suatu karya seni digital, *Graphic Designer* biasanya menggunakan *color pallete* untuk menciptakan kesinambungan antar objek di dalam gambar. *Color Pallete* merupakan sekumpulan warna (yang ada pada gambar) yang membangun suatu gambar. Semakin kompleks *color pallete* pada suatu gambar maka semakin jelas perbedaan warna gambar yang dihasilkan.



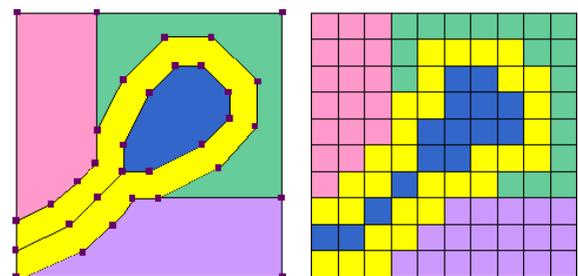
Sumber : Dcoetzee - Own work, CC0,

<https://commons.wikimedia.org/w/index.php?curid=14488053>

Gambar 1. Perbandingan *color pallete* kompleks (kiri) dan *color pallete* 4 warna (kanan)

Untuk membangun gambar dari awal, penentuan palet warna merupakan hal yang mudah karena kita sendiri yang memilih warna yang akan digunakan. Namun, jika pengolahan gambar dilakukan dari suatu foto atau gambar yang sudah tersedia maka biasanya menentukan palet warna menggunakan *feeling* ataupun mengambil beberapa sampel warna dari gambar. Hal tersebut tentu tidak efektif jika banyak gambar yang akan diedit. Akan tetapi, ada cara lain yang lebih mudah untuk menentukan palet warna yaitu *Color Quantization*.

Color Quantization atau Kuantitaasi warna merupakan sebuah proses untuk mengurangi jumlah penggunaan warna pada sebuah gambar. Pengurangan penggunaan warna ditujukan untuk mengurangi ukuran data maupun keterbasan pengolahan gambar visual pada perangkat tertentu. *Color quantization* mudah dilakukan pada gambar berbasis raster (*pixel*) namun susah pada gambar berbasis vektor.



Vector

Raster

Sumber : e-education.psu.edu/natureofgeoinfo

Gambar 2. Perbandingan gambar vektor dan raster

Foto hasil tangkapan kamera poket, DSLR maupun hasil *Photoshop* merupakan gambar berbasis raster. Gambar raster terdiri dari sekumpulan matriks dari *pixel* (susunan paling kecil dari gambar raster). *Pixel* merupakan kode unik yang mengandung nilai warna yang disimpan tiap pesergi atau digital numbers. Setiap satu digital numbers biasanya terdiri dari 24 bit yang mewakili 3 elemen yaitu *red*, *green*, dan *blue* (RGB). Setiap elemennya diwakilkan oleh 8

bit. 8 bit ini menentukan intensitas warna dari masing elemen tersebut, dengan intensitas minimum (00000000=0) menunjukkan warna hitam dan intensitas maksimum (11111111=255) menunjukkan warna putih.

Gambar vektor merupakan gambar berbasis perhitungan matematika (vektor) dimana terdapat simpul-simpul yang akan digabungkan menjadi kurva yang terakhir membentuk objek gambar. Setiap simpul menyimpan informasi warna, ukuran serta arah dari simpul.

Pada makalah ini akan dibahas *color quantization* pada gambar berbasis raster. Karena gambar berbasis raster susunan terkecilnya merupakan *pixel* sehingga bisa dibagi menjadi upa-masalah pada algoritma *divide and conquer*.

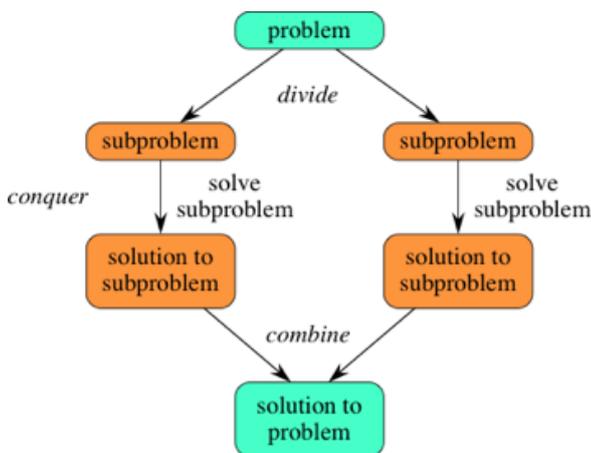
II. DASAR TEORI

A. Divide and Conquer

Algoritma Divide and Conquer adalah metode pemecahan masalah yang cara kerjanya dengan membagi masalah menjadi beberapa upa-masalah (sub-problem) yang lebih kecil, kemudian menyelesaikan masing-masing upa-masalah secara independen, dan akhirnya menggabungkan solusi masing-masing upa masalah menjadi solusi masalah semula.

Algoritma umum Divide and Conquer :

- *Divide* : membagi masalah menjadi beberapa upa masalah yang memiliki kemiripan dengan masalah semula namun berukuran dengan lebih kecil
- *Conquer* : memecahkan masing-masing upa-masalah secara rekursif
- *Combine* : menggabungkan solusi masing-masing upa-masalah sehingga membentuk solusi masalah semula



Sumber : s3.amazonaws.com/ka-cs-algorithms

Gambar 3. Skema umum Algoritma Divide and Conquer

Objek permasalahan yang akan dibagi adalah masukkan atau *instances* berukuran n . Masukkan tersebut mungkin berupa tabel, matriks, eksponen, dan sebagainya, bergantung

pada masalahnya. Tiap-tiap upa-masalah mempunyai karakteristik yang sama (*the same type*) dengan karakteristik masalah asal, sehingga metode *divide and conquer* lebih natural diungkapkan dalam skema rekursif.

Metode *Divide and Conquer* memberikan dua keuntungan. Pertama, metode ini menyediakan pendekatan yang sederhana untuk memecahkan masalah yang secara konseptual sulit, dengan cara mereduksi masalah sampai bisa diselesaikan secara trivial. Kedua, *Divide and Conquer* dapat secara substansial mengurangi biaya (*cost*) komputasi (kompleksitas).

Kompleksitas waktu algoritma *Divide and Conquer* :

$$T(n) = \begin{cases} g(n) & , n \leq n_0 \\ 2T(n/2) + f(n) & , n > n_0 \end{cases}$$

Dengan,

$T(n)$ = waktu komputasi dengan masukkan berukuran n

$g(n)$ = waktu komputasi untuk menyelesaikan upa-masalah yang kecil

$f(n)$ = waktu komputasi untuk menggabungkan solusi masing-masing upa-masalah

B. Rataan dan Variansi

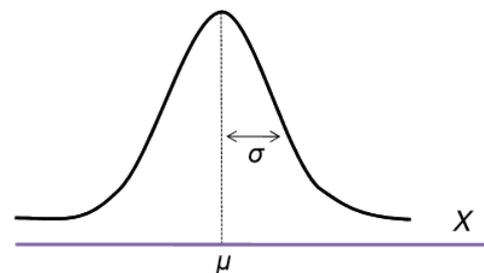
Menghitung rata-ran μ dan variansi σ^2 dari variabel diskrit x adalah :

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\sigma^2 = \frac{\sum (X - \mu)^2}{N}$$

C. Distribusi Normal

Distribusi normal atau dikenal juga dengan *Gaussian Distribution* merupakan distribusi yang penting dalam bidang statistik. Distribusi normal memiliki komponen *mean* (rata-rata) dan simpangan baku. Kurva dari distribusi normal memiliki bentuk yang simetri.



Gambar 4. Kurva Distribusi Normal

Fungsi densitas dari variabel normal X, dengan rata-ran μ dan variansi σ^2 adalah :

$$f(x; \mu, \sigma) = \left(\frac{1}{\sqrt{2\pi}\sigma} \right) e^{-\frac{1}{2} \left[\frac{(x-\mu)}{\sigma} \right]^2}, -\infty < x < \infty$$

Dengan $\pi = 3.14156$ dan $e = 2.7182$.

Sifat-sifat kurva normal :

- Modus, adalah suatu titik yang terletak pada sumbu x di mana kurva mempunyai nilai maksimum, yaitu $x = \mu$
- Kurva berbentuk simetri terhadap sumbu tegak pada $x = \mu$
- Kurva mempunyai titik balik pada $x = \mu \pm \sigma$
- Kurva mendekati sumbu datar secara asimtotik ke dua arah (kiri/kanan) berawal dari μ
- Luas daerah di bawah kurva adalah 1

Distribusi Normal Standard adalah distribusi variabel random normal dengan rata-ran 0 dan variansi 1. Rumus menghitung luas dibawah kurva distribusi normal standard adalah :

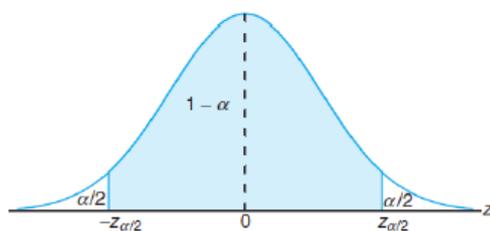
$$Z = \frac{(X - \mu)}{\sigma}$$

Taksiran

Sebuah nilai taksiran dari parameter populasi θ adalah sebuah nilai tunggal ω dari statistik ω .

Sebuah nilai penaksir tidak diharapkan dapat menaksir parameter populasi tanpa kesalahan, misalkan tidak perlu X dapat menaksir μ secara tepat, tetapi diharapkan tidak terlalu jauh dari parameter yang ditaksir.

Penentuan selang taksiran dari μ . Misalkan sampel diambil dari populasi normal, atau jika tidak mempunyai ukuran sampel yang besar. Sesuai dengan teorema limit pusat, diharapkan distribusi sampel X akan diaproksimasi terdistribusi normal dengan rata-ran $\mu_x = \mu$ dan simpangan $\sigma_x = \sigma/\sqrt{n}$.



Karena :

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$$

Maka taksiran interval dapat ditentukan dengan :

$$P\left(\bar{X} - z_{\alpha/2} \frac{\sigma}{\sqrt{n}} < \mu < \bar{X} + z_{\alpha/2} \frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha$$

Salah satu aplikasi dari distribusi normal yaitu teknik *gaussian blur* yang digunakan pada *image-editing tools*. Hasilnya adalah memberikan efek warna dengan mengurangi detail warna pada suatu gambar.



Sumber : IkamusumeFan

Gambar 5. Perbandingan gambar sebelum dan sesudah aplikasi Gaussian Blur

III. ANALISIS DAN IMPLEMENTASI

Gambar raster merupakan array dua dimensi yang disusun oleh *pixel*. *Pixel* merupakan array satu dimensi yang menyimpan informasi warna RGB. Karena merupakan array dua dimensi *pixel* maka kita dapat membagi gambar menjadi upa-maslah array dua dimensi *pixel* yang lebih kecil.

Berikut langkah-langkah penentuan palet warna yang diajukan:

- Bagilah gambar menjadi n bagian (array dua dimensi yang lebih kecil). Semakin besar nilai n maka jangkauan warna yang akan dikuantisasi akan semakin luas. Ini akan memberikan efek pemberian warna yang lebih umum dari suatu gambar. Bila nilai n kecil maka kuantisasi sampel warna akan terlalu detail dan susah terlebih lagi jika kandungan warna berbeda untuk tiap *pixel*
- Selesaikan tiap upa-masalah yaitu dengan menggunakan taksiran distribusi normal. Penggunaan taksiran digunakan untuk menentukan satu warna secara umum yang digunakan oleh upa-masalah gambar. Perhitungan taksiran akan dijelaskan dibawah.
- Gabungkan tiap solusi upa-masalah yang telah diselesaikan ke dalam matriks baru
- Ulangi tahap 1 sampai didapatkan banyaknya palet warna yang diinginkan

Langkah-langkah perhitungan taksiran warna untuk tiap upa-masalah:

- Hitunglah rata-rata untuk tiap penggunaan warna *Red*, *Green*, dan *Blue* pada upa-masalah. Informasi RGB tersebut terdapat pada *pixel*.
- Hitung juga standar deviasi untuk tiap warna *Red*, *Green*, dan *Blue*.
- Tentukan tingkat kepercayaan $1 - \alpha$ (0-100%).

- Gunakan rumus dibawah untuk menghitung selang kepercayaan pada tingkat kepercayaan yang telah ditentukan.

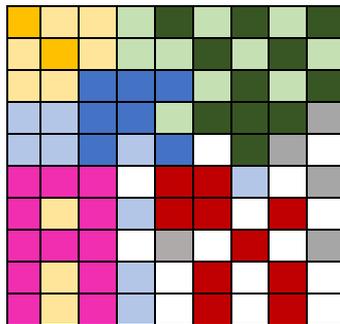
$$P\left(\bar{X} - z_{\alpha/2} \frac{\sigma}{\sqrt{n}} < \mu < \bar{X} + z_{\alpha/2} \frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha$$

Dengan,
 n = jumlah data warna
 x = rata-rata
 σ = standar deviasi

- Setelah perhitungan selesai maka akan didapatkan selang kepercayaan $a < \mu < b$. Dengan a adalah *lower bound* dan b adalah *upper bound*.
- Ambil nilai *lower bound* saja karena nilai *lower bound* merupakan nilai terkecil yang membangun suatu warna.
- Kombinasikan hasil Red, Green, dan Blue untuk mendapatkan warna umum yang digunakan pada gambar
- Lakukan langkah diatas untuk tiap warna pada RGB

Implementasi

Diberikan sebuah potongan gambar berikut.



Gambar 6. Contoh potongan gambar

Dalam pengerjaan contoh diatas, kita dapat mengasumsikan tiap kotak pada tabel adalah tiap *pixel* pada gambar. Selain itu, kita juga mengasumsikan tiap warna memiliki nomor integer tersendiri dengan ketentuan susunan warna harus berurutan sesuai saturasi warna. Hal ini digunakan untuk mempermudah perhitungan RGB pada tiap warna.

1	2	3	4	5	6	7	8	9	10

Tabel 1. Tabel Warna ke Integer

Ubah potongan gambar menjadi matriks integer sesuai tabel 1. Bagi gambar menjadi upa-masalah dengan ukuran 3x3 seperti Gambar 7.

8	3	3	4	9	4	9	4	9
3	8	3	4	4	9	4	9	4
3	3	10	10	10	4	9	4	9
5	5	10	10	4	9	9	9	2
5	5	10	5	10	1	9	2	1
7	7	7	1	6	6	5	1	2
7	3	7	5	6	6	1	6	1

7	7	7	1	2	1	6	1	2
7	8	7	5	1	6	1	6	1

Gambar 7. Konversi ke integer dan pembagian masalah

Selesaikan tiap upa-masalah dengan menggunakan taksiran dengan tingkat kepercayaan 95%.

Tingkat kepercayaan : 95% maka nilai $\alpha = 0.05$. Sehingga, nilai $z_{0,025} = 1.645$.

Penyelesaian upa-masalah 1

8	3	3
3	8	3
3	3	10

$\mu = 4.889$

$\sigma = 2.892$

selang kepercayaan : $3.303350926 < \mu < 6.474426852$

Nilai warna : $3.3033 \approx 3$

Penyelesaian upa-masalah 2

4	9	4
4	4	9
10	10	4

$\mu = 6.444444444$

$\sigma = 2.920235911$

selang kepercayaan : $4.843181753 < \mu < 8.045707136$

Nilai warna : $4.83181 \approx 5$

Penyelesaian upa-masalah 3

9	4	9
4	9	4
9	4	9

$\mu = 6.777777778$

$\sigma = 2.635231383$

selang kepercayaan : $5.332792569 < \mu < 8.222762986$

Nilai warna : $5.3328 \approx 5$

Penyelesaian upa-masalah 4

5	5	10
5	5	10
7	7	7

$\mu = 6.777777778$

$\sigma = 2.048034$

selang kepercayaan : $5.65477231 < \mu < 7.900783246$

Nilai warna : $5.6547 \approx 6$

Penyelesaian upa-masalah 5

10	4	9
5	10	1

1	6	6
---	---	---

$$\mu = 5.777778$$

$$\sigma = 3.456074$$

$$\text{selang kepercayaan : } 3.882697443 < \mu < 7.672858113$$

$$\text{Nilai warna : } 3.8827 \approx 4$$

Penyelesaian upa-masalah 6

9	9	2
9	2	1
5	1	2

$$\mu = 4.444444$$

$$\sigma = 3.609401$$

$$\text{selang kepercayaan : } 2.465289396 < \mu < 6.423599493$$

$$\text{Nilai warna : } 2.465 \approx 2$$

Penyelesaian upa-masalah 7

7	3	7
7	7	7
7	8	7

$$\mu = 6.666667$$

$$\sigma = 1.414214$$

$$\text{selang kepercayaan : } 5.89120623 < \mu < 7.442127103$$

$$\text{Nilai warna : } 5.8912 \approx 6$$

Penyelesaian upa-masalah 8

5	6	6
1	2	1
5	1	6

$$\mu = 3.666667$$

$$\sigma = 2.345208$$

$$\text{selang kepercayaan : } 2.380711013 < \mu < 4.952622321$$

$$\text{Nilai warna : } 2.3807 \approx 2$$

Penyelesaian upa-masalah 9

1	6	1
6	1	2
1	6	1

$$\mu = 2.777778$$

$$\sigma = 2.438123$$

$$\text{selang kepercayaan : } 1.440873589 < \mu < 4.114681966$$

$$\text{Nilai warna : } 1.4408 \approx 1$$

Gabungkan hasil penyelesaian upa-masalah 1-9 kedalam matriks baru.

3	5	5
6	4	2
6	2	1



Gambar 8. Hasil palet warna dengan 6 warna

Hasil diatas menghasilkan palet warna dengan jumlah 6. Untuk menguranginya lagi maka bisa dilakukan dengan membagi menjadi upa-masalah lagi. Karena merupakan matriks ganjil maka pembagian dilakukan dengan cara khusus.

Penyelesaian upa-masalah 10

3	5
6	4

$$\mu = 4.5$$

$$\sigma = 1.290994$$

$$\text{selang kepercayaan : } 3.438157066 < \mu < 5.561842934$$

$$\text{Nilai warna : } 3.4381 \approx 3$$

Penyelesaian upa-masalah 11

5	5
4	2

$$\mu = 4$$

$$\sigma = 1.414214$$

$$\text{selang kepercayaan : } 2.836809345 < \mu < 5.163190655$$

$$\text{Nilai warna : } 2.8362 \approx 3$$

Penyelesaian upa-masalah 12

6	4
6	2

$$\mu = 4.5$$

$$\sigma = 1.914854$$

$$\text{selang kepercayaan : } 2.925032408 < \mu < 6.074967592$$

$$\text{Nilai warna : } 2.92503 \approx 3$$

Penyelesaian upa-masalah 13

4	2
2	1

$$\mu = 2.25$$

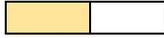
$$\sigma = 1.258306$$

$$\text{selang kepercayaan : } 1.215043529 < \mu < 3.284956471$$

$$\text{Nilai warna : } 1.215 \approx 1$$

Gabungkan hasil penyelesaian upa-masalah 10-13 kedalam matriks baru.

3	3
3	1



Gambar 9. Hasil palet warna dengan 2 warna

IV. KESIMPULAN

Penentuan palet warna menggunakan algoritma diatas menggunakan waktu serta memori yang cukup banyak. Pembacaan isi dari matriks untuk menentukan rata-rata dan variansi membutuhkan kompleksitas $O(n^2)$. Hal tersebut bukan masalah jika gambar berukuran sangat kecil seperti contoh diatas. Namun, biasanya gambar memiliki kepadatan *pixel* yang sangat tinggi. Sehingga cara diatas terbilang kurang efisien meskipun dapat menghasilkan palet warna yang sesuai karena menggunakan taksiran.

Dalam contoh implementasi di Bab 3, hasil warna yang dihasilkan tidak terlalu cocok karena penulis menggunakan konversi ke integer langsung untuk setiap warna tanpa memperhatikan indeks warna RGB masing-masing.

TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa atas waktu yang telah diberikan untuk menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada Bapak Rinaldi Munir dan Ibu Nur Ulfah selaku pengajar yang rela menghabiskan waktu untuk menuangkan ilmu mereka kepada kami. Tidak lupa, penulis juga mengucapkan terima kasih kepada orang tua dan teman-teman saya atas dukungan selama kuliah.

REFERENCES

- [1] Munir, Rinaldi. 2009. *Diktat Kuliah IF3051 Strategi Algoritma*. Bandung: Program Studi Teknik Informatika
- [2] Tim Pengajar. *Slide Kuliah IF 2122 Probabilitas dan Statistik*. ITB.
- [3] "Bitmap Class". Msdn.microsoft.com. Retrieved 5 Mei 2016
- [4] "Patent US6469805 - Post raster-image processing controls for digital color image printing". Google.nl. Retrieved 5 Mei 2016.

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, 6 Mei 2016

Malvin Juanda