PENCARIAN WARNA DASAR DALAM FILE BERISI STRING HEXADESIMAL

Irwan Fathurrahman

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
If18100@students.if.atus itb.ac.id

Abstrak

Tulisan ini berisi cara sederhana dalam pencarian warna dasar pada sebuah foto dengan menggunakan algoritma brute force dengan teknik heuristik. Disini hanya akan diberikan sebuah ide bagaimana penerapannya dan bagaimana implementasinya ke depan, dikarenakan keterbatasan ilmu dan waktu penulis.

Kata kunci: *Warna dasar, brute force, harmoni warna*.

1. PENDAHULUAN

Warna merupakan hal penting bagi seorang artist agar karyanya dapat menarik minat orang lain. Tanpa warna, maka sebuah karya seni akan memiliki suatu kekurangan dalam menjelaskan maksud dari karya seni tersebut.

Namun, banyak seorang yang susah untuk mencari warna apa yang baik untuk karyanya. Ini dapat terlihat saat mengembangkan sebuah *software* yang hanya berbekal seorang *programmer*, yang mana dia harus berusaha untuk membuat interface dari *software* tersebut. Ada yang warnanya terlalu mencolok, ada yang tidak seimbang. Semua ini bermula dari ketidak-harrmonisan warna yang didapat. Padahal, warna sangat penting agar *software* yang dikembangkan menjadi lebih menarik dan mendapat nilai tambah dari pengguna. Maka, penulis mengangkat topik ini akibat banyaknya masalah yang terjadi seberti masalah diatas.

2. Metode

Tulisan ini didapatkan dari referensi yang didapat serta pengalaman yang didapatkan selama pembelajaran menyingkap masalah ini. Selama mencari referensi tersebut, penulis menemukan ide sederhana dalam menyelesaikan masalah ini, walaupun penulis ragu dapat mempublikasikannya.

Pada penggunaannya dapat dicoba tanpa menggunakan sistem dari penulis, namun penulis hanya memberikan sebuah ide algoritma sederhana untuk memudahkan dalam mengimplementasikan metode ini.

3. Penerapan pencarian warna dasar

3. 1 Warna Dasar dalam Harmoni

Gambar 1 Primary Color

Pada dasarnya, sebuah interface terdiri dari beberapa warna dasar yang hanya diatur *hue* maupun *saturationnya*. Dengan pengaturan *hue* maupun *saturation* tersebut, akan didapat warna yang bermacam-macam. Bahkan dengan hanya mencampurkan warna dasar dengan warna dasar yang lain, akan didapat pula warna yang baru. Warna dasar itu adalah merah, kuning dan biru, atau sebutan lainnya adalah *primary color*. Pada dasarnya, semua warna berasal dari ketiga warna tersebut dengan cara mencampurkannya.

Gambar 2 Secondary Color

Namun, jika hanya ketiga warna tersebut yang menjadi dasar dari banyak warna, maka tidak akan ada variasi warna lain yang menarik. Oleh karena itu, maka dari ketiga warna tersebut diciptakan 3 warna lagi, yang disebut *secondary color*, yaitu hijau, oranye dan ungu.

Hijau didapat dengan mencampurkan warna kunig dan biru, orange didapat dengan mencampurkan warna merah dan kuning, dan warna ungu didapat dengan mencampurkan warna biru dan merah.

Gambar 3 Tertiary Color

Dari *secondary color*, akan terbagi lagi warna menjadi 12 warna, yang kemudian akan menjadi warna dasar untuk menciptakan suatu warna yang harmoni, yang disebut *tertiary color*.

Warna inilah yang kemudian menjadi induk atau dasar utama untuk menciptakan warna-warna lain dan menciptakan warna yang harmoni.

Untuk mendapatkan warna-warna harmoni, ada kombinasi-kombinasi dari 12 warna tersebut yang dapat memungkinkan terciptanya warna harmoni.

Dasarnya adalah pada 12 warna ini terdapat 2 kelompok, yaitu :warna panas, yaitu kelompok warna dalam rentang setengah lingkaran di dalam lingkaran warna mulai dari merah hingga kuning. Warna ini menjadi simbol, riang, semangat, marah dsb. Warna panas mengesankan jarak yang dekat. Warna dingin, adalah kelompok warna dalam rentang setengah lingkaran di dalam lingkaran warna mulai dari hijau hingga ungu. Warna ini menjadi simbol kelembutan, sejuk, nyaman dsb. Warna sejuk mengesankan jarak yang jauh.

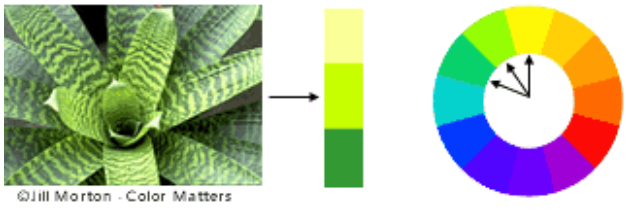
Untuk mendapatkan warna harmoni, dapat juga dengan mengikuti teori-teori yang ada, yaitu :

- *Complement*, yaitu mengambil 2 buah warna yang saling berlawanan arah. Sebagai contoh adalah Merah tua dengan hijau tua.
- *Triad*, yaitu mengambil 3 buah warna yang mana 2 dari 3 warna tersebut merupakan cerminan satu sama lain. Contoh : Kuning, merah dan biru muda, yang mana merah merupakan hasil pencerminan posisi biru muda terhadap sumbu kuning.
- Dan sebagainya

Tertiary color inilah yang akan menjadi **warna dasar** yang akan digunakan untuk menggunakan teori-teori yang ada. Namun, bagaimana cara menentukannya tanpa mencoba setiap warna sesuai dengan tori-teori tersebut?

Alam merupakan suatu ciptaan Tuhan yang memiliki suatu karya yang sangat sempurna. Setiap objek alam memiliki warna yang sangat harmoni yang menarik perhatian kita. Dengan cara melihat suatu objek alam itulah kita akan dapat warna-warna dasar yang harmoni yang akan enak dipandang.

Sebagai contoh di bawah ini :



Gambar 4 Contoh Warna Harmoni Alam

Dari objek suatu bunga tersebut, kita akan mendapatkan suatu deretan warna dasar yang harmoni, yaitu kuning, hijau muda dan hijau tua. Pada warna diatas, telah didapat warna-warna sesuai dengan teori *triad* yang telah dibahas sebelumnya.

Jadi dengan cara itulah kita akan mendapatkan beberapa warna dasar yang harmoni, yaitu dengan mendapatkan warna-warna dari objek alam.

3.2 Ide Mendapatkan Warna Dasar dari Foto

Dengan mencari warna dasar dalam foto objek alam tersebut, kita akan mendapatkan warna-warna harmoni yang enak dipandang untuk *software* kita. Ini dikarenakan objek alam memiliki warna-warna harmoni yang lahir secara alami.

Biasanya, untuk menemukan warna tersebut dari suatu foto objek alam, seorang artis akan menggunakan *tool design* seperti *photoshop* dengan cara mengambil warna yang kira-kira menjadi warna dasar objek. Dari sinilah penulis menemukan ide, bagaimana cara mengambil warna tersebut dengan cepat tanpa perlu bersusah payah menggunakan *tool* berat.

Idenya adalah suatu file gambar dikonversi dulu ke dalam suatu bentuk string panjang dengan merepresentasikan warna (dalam hexadesimal) untuk setiap titik. Dalam tulisan ini, telah diasumsikan bahwa suatu file gambar telah dikonversi menjadi suatu file dengan format, yaitu : resolusi, FOF, string hexadecimal dari warna-warna, EOF. Penulisan belum menemukan metode untuk konversinya karena keterbatasan waktu dan pengetahuan.

Sebagai contoh bentuk file yang telah dikonversi dari file foto :

```
800x600
FOF
cebfbf0400007b2222e40b0b9d4b4b5e3c3c0c0202767767100234899
599F45345cebfbf0400007b2222e40b0b9d4b4b5e3c3c0c0202767767
100234899599F45345cebfbf0400007b2222e40b0b9d4b4b5e3c3c0c0
2027677671002348995 99F45345……cebfbf0400007b2222e
40b0b9d4b4b5e3c3c0c0202767767100234899599F45345cebfbf040
0007b2222e40b0b9d4b4b5e3c3c0c0202767767100234899599F4534
5cebfbf0400007b2222e40b0b9d4b4b5e3c3c0c020276776710023489
9599F45345
EOF
```

Dari awal file ini akan terlihat bahwa resolusi dari foto adalah 800 x 600. Setelah itu akan ada string panjang yang merepresentasikan warna foto. Tadi kita telah mengetahui bahwa panjang dari foto adalah 800. Jadi, akan ada sebanyak 800 warna untuk membentuk baris yang baru. Maksudnya terdapat 800 warna pada baris 1. Warna ke 801 sampai 1600 akan menjadi baris ke-2. Begitu seterusnya sampai baris ke – 600 yang merupakan lebar dari foto.

Lalu bagaimana cara mencari warna-warna dasar yang dimaksud? Yang pertama dilakukan adalah menyimpan isi file yang ke dalam suatu string panjang dengan hanya menyalin file semua isi file tersebut mulai dari FOF sampai dengan EOF. File ini berisi warna dalam bentuk hexadecimal, maka panjang string untuk setiap warna adalah 6 karakter. Sebagai contoh setelah FOF ada warna “cebfbf”. Dengan mencocokkan setiap 6 string dengan data yang telah kita siapkan, maka kita akan mengetahui untuk setiap warna di dalam file apakah warna ini merupakan warna dasar atau bukan.

3.3 Tabel Warna

Satu foto memiliki banyak warna, baik warna dasar, maupun warna yang telah diberikan *hue* maupun *saturation*nya. Untuk itu, kita akan membuat data base yang telah mengelompokkan warna-warna yang mewakili masing-masing warna dasar.

Biru = 0000FF

DodgerBlue	#1E90FF
CornflowerBlue	#6495ED
MediumSlateBlue	#7B68EE
RoyalBlue	#4169E1
Blue	#0000FF
MediumBlue	#0000CD
DarkBlue	#00008B
Navy	#000080
CadetBlue	#5F9EA0
SteelBlue	#4682B4

Biru-hijau / cyan = 000FFF

Aqua	#00FFFF
Cyan	#00FFFF
LightCyan	#E0FFFF
PaleTurquoise	#AFEEEE
Aquamarine	#7FFFD4
Turquoise	#40E0D0
MediumTurquoise	#48D1CC
DarkTurquoise	#00CED1
LightSteelBlue	#B0C4DE
PowderBlue	#B0E0E6
LightBlue	#ADD8E6
SkyBlue	#87CEEB
LightSkyBlue	#87CEFA
DeepSkyBlue	#00BFFF

Hijau = 008000

MediumAquamarine	#66CDAA
DarkSeaGreen	#8FBC8F
LightSeaGreen	#20B2AA
DarkCyan	#008B8B
Teal	#008080
MediumSeaGreen	#3CB371
SeaGreen	#2E8B57
ForestGreen	#228B22
Green	#008000
DarkGreen	#006400
YellowGreen	#9ACD32
OliveDrab	#6B8E23
Olive	#808000
DarkOliveGreen	#556B2F

Kuning-Hijau = ADFF2F

	GreenYellow	#ADFF2F
	Chartreuse	#7FFF00
	LawnGreen	#7CFC00
	Lime	#00FF00
	LimeGreen	#32CD32
	PaleGreen	#98FB98
	LightGreen	#90EE90

Kuning = FFFF00

	Yellow	#FFFF00
	LightYellow	#FFFFE0
	LemonChiffon	#FFFACD
	LightGoldenrodYellow	#FAFAD2
	PapayaWhip	#FFEFD5
	Moccasin	#FFE4B5
	PeachPuff	#FFDAB9
	PaleGoldenrod	#EEE8AA
	Khaki	#F0E68C
	DarkKhaki	#BDB76B

Kuning-orange = FFD700

Orange = FFA500

	DarkOrange	#FF8C00
	Orange	#FFA500

Orange-merah = FF4500

	LightSalmon	#FFA07A
	Coral	#FF7F50
	Tomato	#FF6347
	OrangeRed	#FF4500

Merah = FF0000

	IndianRed	#CD5C5C
	LightCoral	#F08080
	Salmon	#FA8072
	DarkSalmon	#E9967A
	LightSalmon	#FFA07A
	Crimson	#DC143C
	Red	#FF0000
	FireBrick	#B22222
	DarkRed	#8B0000

Merah-ungu = FFC0CB

	Pink	#FFC0CB
	LightPink	#FFB6C1
	HotPink	#FF69B4
	DeepPink	#FF1493
	MediumVioletRed	#C71585
	PaleVioletRed	#DB7093

Ungu = 800080

	Lavender	#E6E6FA
	Thistle	#D8BFD8
	Plum	#DDA0DD
	Violet	#EE82EE
	Orchid	#DA70D6
	Fuchsia	#FF00FF
	Magenta	#FF00FF
	MediumOrchid	#BA55D3
	MediumPurple	#9370DB
	BlueViolet	#8A2BE2
	DarkViolet	#9400D3
	DarkOrchid	#9932CC
	DarkMagenta	#8B008B
	Purple	#800080

Ungu-biru = 4B0082

	Indigo	#4B0082
	SlateBlue	#6A5ACD
	DarkSlateBlue	#483D8B

Gambar 5 Tabel Warna

Untuk data table warna tersebut, kita dapat membuat suatu array, yang mana berisi *hex* warna pada table, *Boolean* dan Warna Dasarnya. Sebagai contoh :

```
Array[0]of TabelWarna = {00008B, false, 0000FF}
```

Ini memiliki arti bahwa dalam senarai ke-0 pada senarai warna memiliki nilai warna 00008B (biru gelap), dengan *Boolean* adalah salah dan memiliki warna dasar 0000FF (Biru).

Dari sinilah struktur data dari table warna nantinya agar sewaktu mencari warna dasar, kita hanya mencocokkan warna yang ditemukan pada string dengan senarai warna.

Apabila warna pada string ditemukan pada struktur data, maka kita akan menyimpan warna dasar apa yang diwakili oleh warna tersebut dan kemudian mengisi *Boolean* dengan true, yang artinya bahwa warna tersebut telah kita temukan.

Namun cara ini tidak akan efisien, karena untuk satu warna dasar terwakili oleh banyak warna. Jika hanya satu senarai diberikan tanda bahwa telah ditemukan, maka warna lain yang memiliki warna dasar yang sama masih tetap akan dibandingkan. Padahal idenya adalah jika satu warna ditemukan, maka warna lainnya yang memiliki warna dasar yang sama tidak akan dibandingkan lagi. Oleh karena itu, terdapat satu senarai lagi yang berisi tentang table warna dasar.

```
Array[0]of TabelWarna = { 00008B,
WarnaDasar[0]}

Array[0]of WarnaDasar = { false, 0000FF}
```

Pada contoh diatas menjelaskan bahwa table warna ke-0 memiliki nilai 00008B dengan warna dasarnya adalah 0000FF dan *boolean*nya *false*. Dengan cara ini, maka kita akan mendapatkan struktur data yang efisien, yang mana jika ada satu warna pada table warna yang diisi *true*, maka warna lain yang memiliki warna dasar yang sama akan terisi sama dengan warna yang dirubah.

Namun, TabelWarna tersebut masih kurang suatu attribute, disaat warna tersebut telah ditemukan salah, maka seharusnya warna tersebut tidak perlu dibandingkan lagi. Oleh karena itu, di TabelWarna harus ditambahkan attribute *Boolean* untuk menentukan apakah warna tersebut masih benar atau telah salah pencocokannya.

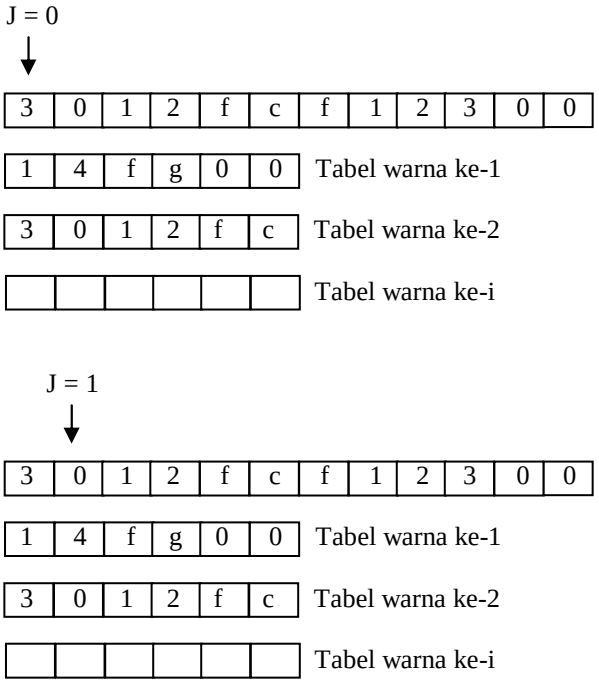
```
Array[0]of TabelWarna = { 00008B, true,
WarnaDasar[0]}

Array[0]of WarnaDasar = { false, 0000FF}
```

Sama seperti sebelumnya, namun untuk yang ini, terdapat tambahan attribute baru pada TabelWarna yang berisi *Boolean* apakah masih boleh untuk dicocokkan atau sudah tidak bisa.

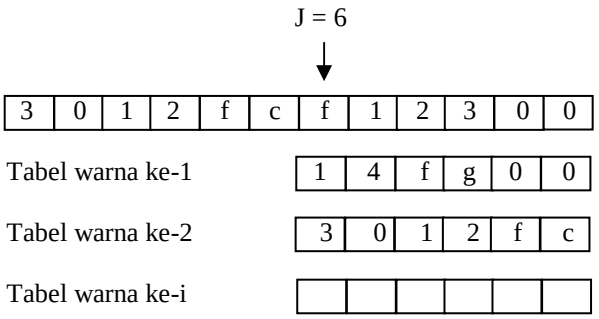
3.4 Pencarian Warna

Selanjutnya adalah mencari warna dasar yang terdapat pada foto. Kita telah memiliki suatu string panjang yang berisi string dari warna-warna pada foto dan kita juga mempunyai tabel warna yang berisi data-data warna yang telah disimpan sebelumnya. Dengan data yang telah dikumpulkan sebelumnya, kita tinggal mencocokkan setiap warna yang terdapat pada string panjang tersebut dengan data tabel warna yang telah kita miliki.

Pada pencarian warna tersebut, pertama akan dilihat setiap karakter di string Warna Foto dengan setiap string di Tabel Warna.



Pencariannya adalah dengan mencocokkan setiap karakter dengan setiap warna pada Tabel Warna. Saat J = 0, maka akan dicocokkan pada string ke-0 dengan string warna ke – 0 pada setiap tabel Warna. Untuk setiap pencocokkan karakter, maka akan dicek apakah karakter ke string j dengan warna ke j pada tabel warna ke i. Jika karakter tidak sama, maka tabel warna ke i yang tidak cocok akan ditandai dengan suatu *boolean* yang berarti tabel warna ini salah dan selanjutnya tidak perlu dicocokkan. Saat j = 5, akan di cek apakah tabel warna tersebu benar atau salah. Jika tabel warna ke i benar, maka akan ditandai warna dasar dari tabel warna ke i yang benar bahwa warna dasar tersebut ada.



Jika j telah mencapai kelipatan 6 , maka pencocokan untuk tabel warna akan dimulai kembali dari warna ke-0. Dan selanjutnya pencocokkan kembali seperti sebelumnya.

Dengan itulah pencarian warna dasar tersebut, yaitu dengan melihat setiap warna dasar, apakah warna dasar tersebut ditemukan atau tidak.

```
Deklarasi :

{ Tabel yang berisi warna yang mewakili }
{ warna dasar }
tabelWarna :
       <
              Warna  : String
              Pencocokkan : Boolean
              WD → warnaDasar {pointer}
       >

{ Tabel yang berisi warna dasar }
{ Tabel ini telah diisi sebelumnya }
{ Ditemukan diisi dengan false }
warnaDasar :
       <
              ditemukan : Boolean
              warnaDasar : String
       >

{ String yang berisi warna-warna pada }
{ foto yang kita cari warna dasarnya }
warnaFoto : String


Function setDefault (input/output WD :
warnaDasar) ← warnaDasar
{ I.S : semua WD.ditemukan berisi true
maupun false }
{ F.S : semua WD.ditemukan untuk semua
warna dasar telah diisi dengan false}

Function setDefaultTabelWarna
(input/output TW : TabelWarna)←
warnaDasar
{ I.S : semua TW.Pencocokan berisi true
maupun false }
{ F.S : semua TW.Pencocokan untuk semua
warna dasar telah diisi dengan true}


Procedure pencocokkan (input/output TW :
tabelWarna, WF : warnaFoto )
{ I.S : Diberikan warnaFoto dan
TabelWarna }
{ F.S : Mengisi Boolean dari warnaDasar}

Function hasil (WD : warnaDasar)
{ I.S : Diberikan warna Dasar yang telah
terisi booleannya }
{ F.S : Menampilkan warna yang merupakan
warna dasar sesuai dengan booleannya}
```

```
Procedure pencocokkan (input/output TW :
tabelWarna, WF : warnaFoto )
{ I.S : Diberikan warnaFoto dan
TabelWarna }
{ F.S : Mengisi Boolean dari warnaDasar}

Deklarasi
j : integer

Algoritma

  j←0
  setDefault(TW.WD)

  for i←0 to WF.size() do
    if j mod 6 = 0 then
      j←0
      setDefaultTabelWarna(TW)
    endif

    for k←0 to TW.size() do
      if TW[k].Warna[j] ≠ WF[i] then
        TW.Pencocokkan = false
      Endif

      if j = 5 then
        TW[k].WD → ditemukan = true
      endif
    endfor
    j←j+1
  endfor
```

4. KESIMPULAN

Brute force merupakan algoritma sederhana yang dapat memecahkan permasalahan apapun. Walaupun brute force merupakan algoritma yang membutuhkan tenaga besar, namun dengan teknik eureka, maka kita akan mengetahui cara yang lebih baik yang dapat mengurangi beban algoritma.

Pencarian string seperti yang dibahas diatas dapat menggunakan algoritma brute force. Walaupun sederhana, namun pencarian yang dilakukan algoritma brute force lebih ”baik” daripada yang lain.

Walaupun pada foto memiliki banyak warna, namun ada beberapa warna yang dapat dijadikan patokan untuk mencari warna dasar yang telah dikelompokkan pada tulisan ini.

REFERENSI

www.W3school.com

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, 7 Desember 2010

Irwan Fathurrahman - 13508100