

IMPLEMENTASI COLOR RECOGNITION DENGAN METODE DYNAMIC COLOR THRESHOLDING UNTUK SISTEM INFORMASI BARANG

by Ali Mahmudi

Submission date: 21-Oct-2020 11:13AM (UTC+0700)

Submission ID: 1421756871

File name: paper_color_threshold-min.pdf (238.12K)

Word count: 2444

Character count: 13939

IMPLEMENTASI *COLOR RECOGNITION* DENGAN METODE *DYNAMIC COLOR THRESHOLDING* UNTUK SISTEM INFORMASI BARANG

Ali Mahmudi¹, Sidik Noertjahjono², Donny Eko Suprayogo³

¹*Amahmudi@hotmail.com*, ²*Sidik@fti.itn.ac.id*, ³*Donny.E.Suprayogo@gmail.com*
Teknik Informatika, Institut Teknologi Nasional Malang, Jln Raya Karanglo km 2, Malang

Abstrak

Penggunaan kombinasi warna sebagai karakter untuk membantu kerahasiaan dan mempercepat pemrosesan data. Penelitian ini menggunakan webcam sebagai input, kemudian diproses oleh komputer dan berikutnya dideteksi karakter yang mewakili kombinasi warna tersebut. Aplikasi ini menggunakan kombinasi dari 3 warna dasar yaitu Red, Green dan Blue.

Pada penelitian ini, proses mendeteksi warna menggunakan metode Color Thresholding. Metode ini menggunakan ambang batas nilai intensitas, untuk menentukan sebuah warna. Warna Red jika komponen warna $Red > 200$, $Green < 64$ dan $Blue < 64$. Warna Green jika komponen warna $Red < 64$, $Green > 200$ dan $Blue < 64$. Warna Blue jika $Red < 64$, $Green < 64$ dan $Blue > 200$. Penelitian ini menggunakan 3 posisi dengan 3 warna dasar, sehingga menghasilkan $3^3 = 27$ kombinasi. Dari 27 kombinasi, yang dipakai adalah 10 kombinasi dan terhubung pada sistem informasi dengan database Oracle [Utami (2010), Setiaji (2012)].

Pengujian tahap pertama dengan 5 sampel (dari 45 karakter) yang dibaca secara berurutan hasilnya 100% berhasil terbaca. Pengujian tahap kedua 5 sampel (dari 180 karakter) dibaca secara acak selama 20 kali terjadi kesalahan 1 karakter sehingga persentase keberhasilannya adalah 98,875%.

Kata kunci : Color Thresholding, Color Recognition, Sistem Informasi barang

1. Pendahuluan

Dengan menganalisa tiap piksel sebuah citra digital maka bisa dibuat sebuah kode untuk mempercepat proses pembacaan [Gonzales & Woods (2002)]. Kode yang tersambung dengan database lebih mempercepat akses informasi dari kode yang telah dikenali. Suatu sistem informasi yang didukung oleh *Computer Vision* [Susanto (2012)] sebagai identifikasi kode akan meningkatkan efisiensi pekerjaan. Selain itu database yang dilengkapi dengan sistem client- server akan mempermudah distribusi data dan menghindari redundansi data antara client dan server.

Metode yang digunakan dalam penelitian ini adalah metode *Dynamic Color Threshold*. Metode ini untuk mendeteksi warna dengan memberikan ambang batas nilai RGB untuk menentukan nilai intensitas warna yang ada pada sebuah citra. Penelitian ini mengambil studi kasus pada Gudang PLTA Sutami karena pencatatan barang masih menggunakan cara manual.

2. Landasan Teori

2.1 Citra RGB

RGB adalah singkatan dari *Red - Green - Blue* adalah model warna pencahayaan (*additive color mode*) dipakai untuk "*input devices*" seperti scanner dan "*output devices*" seperti display monitor. Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar ($RGB = Red\ Green\ Blue$). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 255 level. Berarti setiap piksel mempunyai kombinasi warna sebanyak $2^8 \cdot 2^8 \cdot 2^8 = 2^{24} = 16$ juta warna lebih [Anonim (2012)].

2.2 Threshold RGB

Untuk warna-warna dasar, nilai RGB cukup efektif dalam melakukan deteksi. *Threshold* RGB ini sangat rentan terhadap perubahan kestabilan cahaya. Aplikasi untuk menentukan nilai *Threshold* ditunjukkan pada gambar 1.

Untuk menentukan warna RGB dengan metode thresholding
IF $r > 200$ and $g < 64$ and $b < 64$ THEN
 $warna = Red$
 Nilai – nilai yang berwarna Red adalah nilai thresholding yang menjadi batasan dari penentuan warna

Gambar 1. Perhitungan RGB

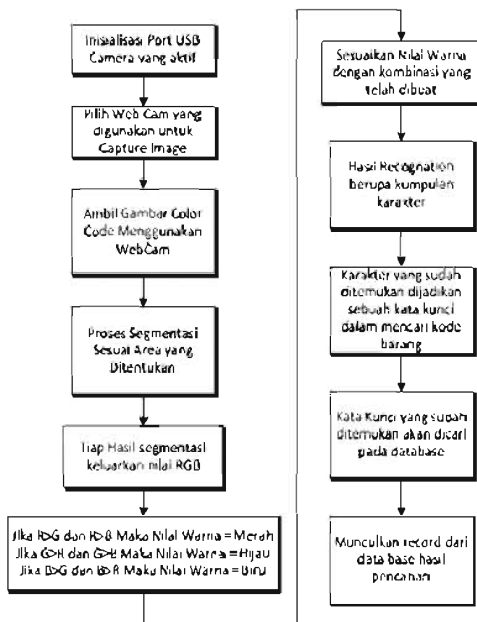
3 Untuk menentukan warna dari suatu citra, dari data warna-warna tersebut diambil rata-rata dari setiap elemen warna. *Thresholding* dilakukan dengan menghitung setiap rata-rata elemen warna. Apabila nilai di antara ketiga warna tersebut ada yang lebih dominan maka nilai dari piksel itu adalah warna yang dominan.

3. Perancangan Dan Implementasi

3.1 Analisa Kebutuhan

Sistem ini terbagi dalam beberapa proses yang saling berhubungan. Salah satu proses yang akan diproses yaitu proses *recognition* yaitu pengenalan *Color Code* sesuai dengan komposisi warna yang ditangkap oleh kamera. Gambar 2 adalah Blok diagram proses *recognition*.

DIAGRAM BLOK COLOR RECOGNITION



Gambar 2. Diagram Blok Color Recognition

Pendeteksian warna dilakukan dengan metode *color Thresholding*. Kombinasi warna yang dipakai untuk menentukan sebuah karakter ada 3 warna yaitu Red, Green, Blue. Tabel 1 berisi tentang syarat nilai intensitas RGB pada sebuah image yang dianggap memiliki warna Red, Green dan Blue. Sebelum dilakukan identifikasi warna dilakukan dulu proses segmentasi. Nilai RGB yang tertera pada tabel adalah nilai batas yang dijadikan *ColorThreshold*.

Tabel 1. Penentuan nilai warna

	Intensitas Nilai RGB		
Red	$R > 180$	$G < 64$	$B < 64$
Green	$R < 64$	$R > 180$	$B < 64$
Blue	$R < 64$	$G < 64$	$R > 180$

3.2 Segmentasi Gambar

Proses segmentasi dilakukan untuk memilih citra yang akan diproses, karena tidak semua citra hasil *capture* akan diproses untuk mengurangi beban memori yang akan dipakai. Proses segmentasi citra ini akan dilakukan sebanyak 30 kali, sehingga akan dihasilkan 30 gambar baru dari hasil segmentasi. Tiga segmentasi pertama akan dilakukan untuk mengunci citra yang akan di proses. Kemudian dilanjutkan dengan segmentasi kedua yaitu 27 kotak warna yang akan diterjemahkan ke dalam bilangan desimal. Hasil segmentasi akan ditampilkan ke dalam komponen Image milik Delphi. Ukuran citra yang dihasilkan dari 30x30 piksel untuk segmentasi pengunci ukurannya lebih besar yaitu 90x40 piksel. Komponen timer digunakan untuk mengatur interval waktu proses *capture*. Gambar 3 adalah gambaran dari proses segmentasi yang dilakukan.



Gambar 3. Segmentasi Color Code

Pada gambar 3 tanda P1, P2, P3 adalah warna yang dijadikan pengunci ketika warna itu terdeteksi

maka program akan melakukan capture citra melalui webcam. Tabel 2 adalah pola pembacaan warna dari hasil segmentasi.

3.3 Kombinasi Warna

Untuk warna yang dibaca adalah warna RGB, untuk menentukan suatu nilai maka dibutuhkan 3 kombinasi warna, Misalnya saja angka 1 di tentukan oleh kombinasi warna Red, Blue, Green. Tabel 3 adalah tabel kombinasi warna sebagai basis pengetahuan dari proses color recognition yang dilakukan pada penelitian ini.

Tabel 2. Pola Urutan Pembacaan Color Code

warna hasil segmentasi	Nomor Urut Karakter
P1	Tanda Red
P2	Tanda Green
P3	Tanda Blue
K1.K2.K3	1
K4.K5.K6	2
K7.K8.K9	3
K10.K11.K12	4
K13.K14.K15	5
K16.K17.K18	6
K19.K20.K21	7
K22.K23.K24	8
25.K26.K27	9

Tabel 3. Kombinasi Color Code

Kombinasi Warna	Warna	Karakter yang dihasilkan
	Red, Green, Blue	0
	Red, Blue, Green	1
	Green, Red, Blue	2
	Green, Blue, Red	3
	Blue, Red, Green	4
	Blue, Green, Red	5
	Red, Green, Red	6
	Green, Red, Green	7
	Blue, Red, Blue	8
	Red, Blue, Red	9

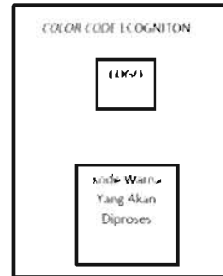
3.4 Desain Color Code

Color Code yang digunakan terbuat dari mika. Warna hitam digunakan sebagai background warna dan posisi warna yang digunakan adalah posisi portrait. Kombinasi warna yang akan dibaca ada pada bagian bawah Color Code. Desain belakang Color Code hanya berisi logo dari Color Code. Desain color code ditunjukkan oleh gambar 4.

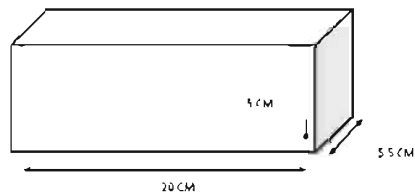
Perancangan Box Pembacaan Kode Warna

Proses pertama yang dilakukan dalam pembacaan kode warna adalah menempatkan kartu pada sebuah box yang sudah disediakan sehingga pembacaan akan lebih akurat. Pada box ini sudah ditentukan berapa jarak antara kamera dengan kartu kode warna yang akan dibaca. Gambar 4, 5 dan 6 adalah desain box untuk membaca Color Code.

Warna yang dipilih adalah warna hitam karena mencegah masuknya cahaya yang akan mengganggu proses pembacaan Color Code karena dengan warna hitam kondisi di dalam box akan gelap dan satu-satunya sumber cahaya ada pada lampu led.



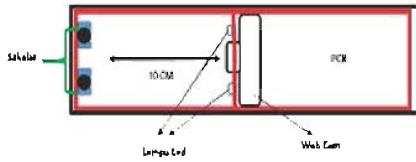
Gambar 4. Tampilan Depan Color Code



Gambar 5. Tampilan Belakang Color Code



Gambar 6. Box Pembaca Kartu



Gambar 7. Desain Box Pembaca Kartu

Seperti ditunjukkan pada gambar 7, di dalam box terdapat sebuah webcam untuk menangkap gambar Box ini juga memiliki sebuah switch dan USB terminal untuk menghubungkan lampu led yang ada pada box dengan webcam. Ketika kartu menempel pada dasar box, maka kartu secara otomatis menekan switch menyalakan lampu led dan cahaya akan mengarah kepada kartu kode warna. Lampu led ini berfungsi untuk memberi cahaya yang mantab.



Gambar 8. Tampilan Box Pembaca Kartu

Gambar 8 adalah hasil jadi dari desain box yang dibuat dari mika tebal dengan ketebalan 1 cm. Pada bagian belakang webcam terdapat kabel webcam dan USB hub yang menghubungkan antara camera webcam dengan lampu led. Lampu led yang dipasang mengambil catu daya lewat USB.



Gambar 9. USB Hub Untuk Menyambungkan Antara Web Cam Dengan Lampu Led

Pada USB hub yang diambil oleh lampu led hanya jalur dari tegangan USB saja sedangkan untuk jalur data USB tidak dipakai. Karena pada USB ada 4 jalur yaitu 2 jalur untuk tegangan dan 2 jalur untuk lalu lintas data.

4. Hasil Dan Pengujian

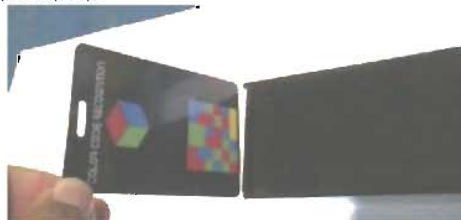
4.1 Pembacaan Kode Warna

Hasil dari pembacaan kode adalah sebuah bilangan desimal yang mewakili sebuah kode barang. Kode barang yang terbaca merupakan kata kunci untuk mencari informasi data barang. Dalam proses capture kode warna, hal pertama yang diproses adalah pemilihan jenis kamera yang aktif. Dalam combo box akan ditampilkan semua hardware kamera yang dideteksi oleh komputer. Ketika salah satu kamera dipilih, maka kamera tersebut akan aktif. Dengan menggunakan komponen DSPack, gambar yang ditangkap oleh kamera akan ditampilkan pada komponen Video Windows.



Gambar 10. Form Pembacaan Color Code

Setelah memilih kamera yang aktif langkah selanjutnya adalah memasukkan Color Code pada Box Reader yang telah dibuat (gambar 11). Ketika ada tanda bunyi, maka pembacaan Color Code telah berhasil dilakukan dan kartu sudah bisa diambil dari Box Reader.



Gambar 11. Memasukkan Color Code Ke Dalam Box Reader

4.2 Integrasi Sistem Informasi Gudang

Setelah mendapatkan kata kunci yang terdapat pada kode warna maka akan dimasukkan ke dalam data base dan akan ditampilkan informasi mengenai barang yang dimaksudkan.

4.2.1 Form Login

Form yang akan ditampilkan pertama kali adalah form login (gambar 12). Form validasi ini adalah sebagai penyeleksi user yang masuk karena

tidak semua user berhak merubah data. Ada 2 level user. User level 2 dapat membuka semua form dan mengakses tombol simpan, edit, hapus dan cari. Sedangkan user level 1 hanya dapat melihat beberapa form dan tidak mempunyai hak untuk melakukan simpan, edit, hapus, cari hanya bisa melihat beberapa laporan saja.

4.2.2 Menu Utama

Pada tampilan awal menu akan tampak beberapa menu utama dimana setiap menu akan berhubungan dalam mengolah database (gambar 13). Pilihan untuk melakukan pembacaan warna ada pada menu Data.



Gambar 12. Form Login Untuk Validasi Hak Akses



Gambar 13. Menu Utama Program

4.2.3 Form Data Barang

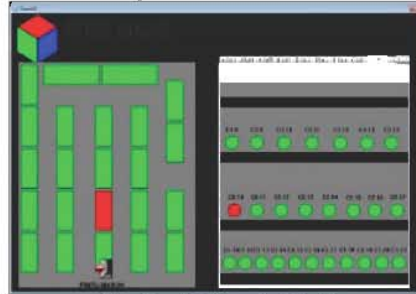
Pada form data barang (gambar 14), user dapat melakukan simpan, edit, hapus dan cari; hanya untuk user level 2. Pada form ini, user bisa melihat gambar barang serta posisi dari barang yang dipilih dengan cara mengklik tombol lokasi. Pada pojok sebelah kanan terlihat juga gambar dari sebuah barang yang dipilih. Aplikasi ini diberi tambahan gambar barang untuk barang-barang yang masih dalam kardus. Selain itu juga terdapat ratusan bentuk barang sehingga tidak bisa menghafal kan bentuk barangnya satu per satu.



Gambar 14. Form pengolah data barang

4.2.4 Form Denah Lokasi

Form ini adalah salah satu fitur aplikasi untuk menunjukkan barang yang dicari pada sebuah peta planogram gudang. Disini barang yang dipilih akan ditunjukkan dengan warna titik yang berkedip-kedip, dan titik yang sebelumnya berwarna Green akan berubah menjadi Red.



Gambar 15. Denah lokasi

4.2.5 Form Penerimaan PO

Form ini digunakan untuk melihat laporan penerimaan yang terjadi. Pada form ini, seperti ditunjukkan pada gambar 16, dapat melihat penerimaan barang berdasarkan Nomor PO, Nama Barang dan Stock Code. Kemudian user tinggal memilih preview maka aplikasi akan menampilkan semua data penerimaan sesuai dengan jenis yang dipilih.



Gambar 16. Form pengolah PO

4.3 Pengujian

Pengujian pertama dilakukan dengan cara pembacaan pada beberapa sampel kode warna yang telah dibuat. Dalam pengujian ini dilakukan 10 kali pengujian kartu dengan kode yang berbeda-beda dan dilakukan dalam beberapa kali perulangan. Dengan pengujian ini bisa diperoleh berapa tingkat akurasi pembacaan kode warna. Hasil pengujian akan ditampilkan pada tabel 4.

Tabel 4. Hasil Pengujian Tahap 1

Sample	Nilai Kode Warna	Hasil Pembacaan	Presentase keberhasilan
1	000746008	000746008	100%

2	000791913	000791913	100%
3	000805481	000805481	100%
4	123456780	123456780	100%
5	000327304	000327304	100%
Persentase Keberhasilan			100%

Pengujian kedua dengan menggunakan sampel acak dan dilakukan dalam beberapa kali pembacaan. Hal ini bertujuan untuk mengetahui akurasi jika menggunakan data yang sama dan acak secara berkala apakah akan mempengaruhi tingkat akurasi atau tidak. Hal ini akan dilakukan sebanyak 20 kali pada pengujian ini akan mendapatkan beberapa sampel yang memiliki nilai Sama.

Tabel 5. Hasil Pengujian Tahap 2

Sample	Nilai Kode Warna	Hasil Pembacaan	Persentase keberhasilan
1	000327304	000327304	100%
2	000746008	000746008	100%
3	000327304	000327304	100%
4	000791913	000791913	100%
5	000805481	000805481	100%
6	123456780	123456780	100%
7	000746008	000746008	100%
8	000327304	000327304	100%
9	000791913	000791913	100%
10	000746008	000746008	100%
11	123456780	123456780	100%
12	000746008	000746008	100%
13	000791913	000791913	100%
14	123456780	123456780	100%
15	000805481	000806481	98,875%
16	000746008	000746008	100%
17	000805481	000805481	100%
18	000791913	000791913	100%
19	000746008	000746008	100%
20	000805481	000805481	100%
Persentase Keberhasilan			99,944%

Pengujian tahap I seperti ditunjukkan pada Tabel 4 menunjukkan keberhasilan pembacaan 100%. Pada pengujian kedua seperti pada Tabel 5 terdapat salah satu kode yang salah membaca kode yaitu pada *Color Code* 000805481 salah pada angka 5 menjadi angka 6 sehingga hasil pembacaannya menjadi 000806481. Kesalahan terjadi mungkin dikarenakan

adanya cahaya luar yang masuk, atau faktor posisi kartu saat dimasukkan pada color reader.

5. Kesimpulan

1. Setiap warna dapat mewakili sebuah kode sehingga dengan warna dapat dibuat sebuah kata kunci khusus untuk menghindari redundansi data.
2. Pendeteksian warna ini dinyatakan 100% berhasil walaupun dalam pengujian dengan sampel secara acak peluang keberhasilannya 98,875%.
3. Dalam melakukan deteksi warna, intensitas cahaya sekitar akan mempengaruhi nilai pembacaan intensitas warna dasar RGB.

Daftar Pustaka:

- Achmad, A & Firdausy, K, 2005, *Teknik Pengolahan Citra Digital Menggunakan Delphi*, Ardi Publishing.
- Anonim, 2012, *Distributed Database Using Oracle.pdf* (diakses tanggal 02 Maret 2012).
- Ferdinand, M, 2012, *SI*, [pdf], (<http://april-si.comuf.com/SI.pdf>, diakses tanggal 2 Januari 2013).
- Gonzalez, R dan and Woods, R, 2002, *Digital Image Processing*, Addison-Wesley Publishing.
- Iswanto, N, et al " *Desain dan Implementasi Color Code Untuk Verifikasi Nomor Kendaraan Bermotor pada Sistem Parkir*", 2012, Program Studi Telekomunikasi, Institut Teknologi Telkom.
- Sembiring, A, 2012, *Pembentukan Citra*, [pdf], (http://asanisembiring.files.wordpress.com/2012/02/bab-2_pembentukan-citra.pdf, diakses tanggal 19 November 2013)
- Setiaji, H, 2012, *Distributed Database Using Oracle*, [pdf], (<http://www.slideshare.net/harisetiaji31/distributed-database-using-oracle>, diakses tanggal 2 Maret 2012)
- Sirait Hasanuddin, 2012, *TEORI DASAR CITRA DIGITAL*, [pdf], (<http://sirait.files.wordpress.com/2009/02/citra-digital-13.pdf>, diakses tanggal 19 November 2012)
- Susanto, B, 2012, *Membangun Sistem Basis Data Dengan ORACLE XE*, Andi Publisher.
- Sutoyo, et al, 2009, *Teori Pengolahan Citra Digital*, Andi Publisher.
- Tata Laksana Gudang PLTA SUTAMI, 2010.
- Utami, J, 2010, *MATERI DELPHI*, [pdf], (<http://jutami.staff.gunadarma.ac.id/Downloads/files/29491/Sejarah+Delphi.doc>, diakses tanggal 15 Januari 2013)

IMPLEMENTASI COLOR RECOGNITION DENGAN METODE DYNAMIC COLOR THRESHOLDING UNTUK SISTEM INFORMASI BARANG

ORIGINALITY REPORT

6%

SIMILARITY INDEX

6%

INTERNET SOURCES

2%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Universitas Brawijaya

Student Paper

3%

2

jurnal.upnyk.ac.id

Internet Source

2%

3

dokumen.tips

Internet Source

2%

Exclude quotes On

Exclude bibliography On

Exclude matches < 2%