PENERAPAN METODE *TEMPLATE MATCHING* UNTUK MENGHITUNG NILAI HAMBATAN RESISTOR

Dedi Sukardi Hardhiana[1], Galih Hermawan[2]
[1,2] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112-114 Bandung, Jawa Barat 40132
E-mail: dedihardhiana94@gmail.com[1], galih.hermawan@email.unikom.ac.id[2]

ABSTRAK

Resistor merupakan suatu komponen dasar elektronika yang berfungsi untuk membatasi jumlah arus dalam suatu rangkaian. Tipe resistor yang umum berbentuk tabung dan terdapat beberapa cincin warna. Kode warna pada resistor menunjukan nilai suatu hambatan pada resistor. Berdasarkan pada hasil penelitian sebelumnya, yang memiliki presentase keberhasilan dalam perhitungan nilai sebesar 57%, untuk salah dikenali 30%, dan tidak dikenali 13%. Maka dari itu, diperlukan sebuah penelitian untuk membuat sebuah aplikasi yang menghitung nilai dengan menggunakan pengolahan citra digital. Metode *template matching* merupakan salah satu metode terapan dari teknik konvolusi. Prinsip metode ini adalah membandingkan citra objek yang akan dikenali dengan *image template* yang ada. Bila citra objek memiliki nilai kemiripan tinggi maka image objek akan dikenali. Untuk mengimplementasikan metode *template matching* diperlukan operasi pengolahan citra yaitu klasifikasi warna HSV (*Hue Saturation Value*) yang digunakan untuk mendeteksi warna pada resistor. Setelah melakukan pengujian sistem dengan metode *Black Box* dapat ditarik kesimpulan bahwa program berfungsi dengan benar. Pada pendeteksian objek memiliki tingkat akurasi 89% dan untuk perhitungan nilai hambatan resistor memiliki tingkat akurasi 69%.

Kata kunci: resistor, *template matching, hue, saturation, value* , nilai hambatan, pengenalan pola.

1. PENDAHULUAN

Resistor adalah komponen elektronika yang berfungsi untuk menghambat arus yang masuk pada rangkaian elektronika. Tipe resistor yang umum berbentuk tabung dan terdapat beberapa cincin warna. Cincin warna pada resistor merupakan kode standar manufaktur yang dikeluarkan oleh EIA (*Electronic Industries Association*) [1].

Kode warna pada resistor menunjukan nilai suatu hambatan pada resistor. Pembacaan kode warna memiliki beberapa warna di setiap cincin. Sehingga memiliki beberapa kombinasi antara warna cincin dan jumlah cincin. Kesalahan pembacaan kode warna mengakibatkan salah dalam menentukan nilai hambatan resistor. Nilai hambatan yang tidak sesuai dengan rangkaian akan mengakibatkan kerusakan pada rangkaian elektronika. Berdasarkan pada hasil penelitian sebelumnya, yang memiliki presentase keberhasilan dalam perhitungan nilai sebesar 57%, untuk salah dikenali 30%, dan tidak dikenali 13% [2]. Maka dari itu, diperlukan sebuah penilitian untuk membuat sebuah aplikasi yang menghitung nilai dengan menggunakan pengolahan citra digital.

Pengolahan citra merupakan teknik manipulasi citra secara digital yang khususnya menggunakan komputer. Agar mudah diinterpretasi oleh manusia dan komputer, maka pengolahan citra digital harus dilakukan dengan berbagai metode untuk mencapai citra yang sesuai [3]. Pengolahan citra digital yang menggunakan metode pencocokan pola untuk mendeteksi salah satunya yaitu dengan metode *template matching*.

Metode *template matching* adalah salah satu metode terapan dari teknik konvolusi. Penerapan metode *template matching* relatif mudah diimplementasikan pada teknik pengolahan citra digital. Prinsip metode ini adalah membandingkan citra objek yang akan dikenali dengan citra *template* yang ada. Bila image objek memiliki nilai kemiripan tinggi maka citra objek akan terdeteksi. Sebaliknya, bila citra objek memiliki nilai kemiripan yang rendah maka citra objek dikategorikan sebagai objek yang tidak diketahui [2].

Selanjutnya untuk dapat mengimplementasikan metode ini maka perlu beberapa operasi pengolahan citra digital salah satunya dengan klasifikasi warna HSV. Klasifikasi warna HSV digunakan untuk mendeteksi kode warna pada resistor yang akan diuji. Keuntungannya yaitu sederhana dalam pemrograman dan prosesnya cepat.

Berdasarkan pemaparan di atas, penelitian dimaksudkan untuk menerapkan metode *template matching* dan klasifikasi warna HSV untuk menghitung nilai hambatan resistor pada suatu rangkaian.

2. ISI PENELITIAN

2.1 Tinjauan Pustaka

2.1.1 Resistor

Resistor adalah komponen elektronika yang berfungsi untuk menghambat arus yang masuk pada rangkaian elektronika. Satuan nilai dari resistor adalah ohm (Ω) [1].

2.1.2 *Template matching*

Template matching merupakan salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokan setiap bagian dari suatu citra dengan citra yang menjadi *template* [7]. Pada dasarnya proses dilakukan dengan cara memasukan citra *template* tertentu kemudian dibandingkan dengan citra pada database.

2.1.3 *Normalized Cross-correlation*

Normalisasi merupakan salah satu proses dalam tahap *preprocessing* yang dilakukan sebelum masuk ke proses pendeteksian objek. Proses yang digunakan pada tahap normalisasi ini mengggunakan konsep normalisasi sesuai dengan metode *Normalized Coss Corelation*, dimana dalam tahapan ini meliputi, menghitung rata-rata dan standar deviasi untuk citra uji dan citra template. Persamaan untuk menghitung rata-rata sebagai berikut :

$$\bar{x} = \frac{x1+x2+x3+\cdots+xn}{n} \quad (2.1)$$

Sedangkan persamaan untuk menghitung standar deviasi sebagai berikut :

$$\hat{f} = \frac{\mathrm{f}-\bar{\mathrm{f}}}{\sqrt{\Sigma(\mathrm{f}-\bar{\mathrm{f}})^2}} \quad \hat{\mathrm{g}} = \frac{\mathrm{g}-\bar{g}}{\sqrt{\Sigma(\mathrm{g}-\bar{\mathrm{g}})^2}} \quad (2.2)$$

2.1.4 Model Warna HSV

Pada pengolahan warna gambar, ada bermacam model warna. Model RGB (*red green blue*) merupakan model yang banyak digunakan. Pada model ini untuk merepresentasikan gambar menggunakan 3 buah komponen warna tersebut. Selain RGB terdapat juga model warna HSV dimana model ini terdapat 3 komponen yaitu *hue*, *saturation*, dan *value*. *Hue* adalah salah satu ukuran panjang gelombang yang terdapat pada warna dominan yang diterima oleh penglihatan, sedangkan *saturation* adalah ukuran banyaknya cahaya putih yang bercampur pada *hue.*

$$R' = \frac{R}{255} \quad (2.3)$$

$$G' = \frac{G}{255} \quad (2.4)$$

$$B' = \frac{B}{255} \quad (2.5)$$

$$H = \begin{cases} 0, & jika\ max = min \\ 60° \ x \left(\frac{(G-B)}{max-min}\ mod\ 6\right) \\ 60° \ x \left(\frac{(B-R)}{max-min} + 2\right) \\ 60° \ x \left(\frac{(R-G)}{max-min} + 4\right) \end{cases} \quad (2.6)$$

$$S = \begin{cases} 0, & jika\ max = min \\ \frac{max-min}{v} \end{cases} \quad (2.7)$$

$$V = max \quad (2.8)$$

2.1.5 Kuantisasi Warna HSV

Kuantisasi warna adalah proses pengurangan jumlah warna yang mereprentasi suatu citra. Salah satu kombinasi dimensi yang baik dari segi performa serta beban komputasi yang ringan dan sering digunakan yaitu 72 dimensi (8 dimensi untuk komponen *hue*, 3 dimensi untuk komponen *saturation*, dan 3 dimensi untuk komponen *value*).

$$H = \begin{Bmatrix} 0\ jika\ h\ \in [316-20] \\ 1\ jika\ h\ \in [21-40] \\ 2\ jika\ h\ \in [41-75] \\ 3\ jika\ h\ \in [76-155] \\ 4\ jika\ h\ \in [156-190] \\ 5\ jika\ h\ \in [191-270] \\ 6\ jika\ h\ \in [271-295] \\ 7\ jika\ h\ \in [296-315] \end{Bmatrix} \quad (2.9)$$

$$S = \begin{Bmatrix} 0\ jika\ s\ \in [0-0.2] \\ 1\ jika\ s\ \in [0.2-0.7] \\ 2\ jika\ s\ \in [0.7-1] \end{Bmatrix} \quad (2.10)$$

$$V = \begin{Bmatrix} 0\ jika\ v\ \in [0-0.2] \\ 1\ jika\ v\ \in [0.2-0.7] \\ 2\ jika\ v\ \in [0.7-1] \end{Bmatrix} \quad (2.11)$$

Setelah mendapatkan nilai 72 dimensi, nilai HSV di normalisasi menjadi 1 vektor dengan persamaan

$$G = 9H + 3S + V \quad (2.12)$$

2.1.6 Konvolusi

Konvolusi adalah sebuah proses dimana citra dimanipulasi dengan eksternal mask (*template*) untuk menghasilkan citra yang baru. Dimana *template* dinyatakan dalam bentuk matriks yang ukurannya lebih kecil dari ukuran citranya [8]. Konvolusi pada operasi 2 dimensi dapat didefinisikan pada persamaan berikut:

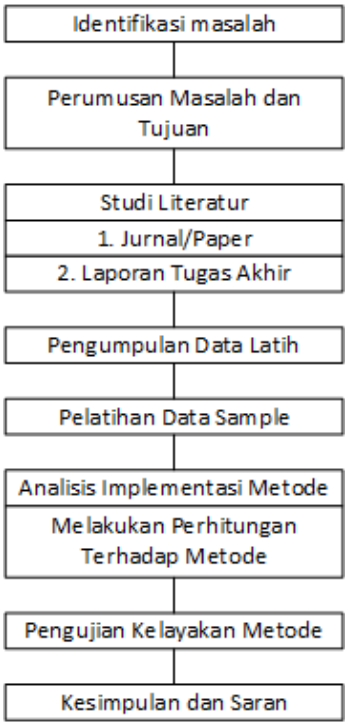
$$I'(\hat{\mathrm{g}}, \hat{\mathrm{f}}) = \sum_{i=-n}^{n} \sum_{j=-m-1}^{m} h(i,j) I(n,m) \quad (2.13)$$

Operasi konvolusi dilakukan dengan cara menggeser mask *pixel* per *pixel* pada citra. Dimulai dari *pixel* awal (0,0) ke *pixel* (0,1) dan seterusnya hingga akhir dari pixel. Hasil konvolusi akan disimpan dalam matriks baru.

2.2 Analisa dan Implementasi

2.2.1 Metode Pengumpulan Data

Dalam penulisan laporan tugas akhir ini, metode penelitian ini dibagi menjadi 2 tahap yaitu tahap pengumpulan data dan pembangunan perangkat lunak.



Gambar 1. Skema Penelitian

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut:

a. Studi Literatur

Metode pengumpulan data yang berkaitan dengan penelitian yaitu berupa buku-buku, *paper*, jurnal yang berkaitan dengan *template matching* dan HSV.

b. Pengumpulan Sample

Pada tahap ini peneliti mengumpulkan data sample berupa gambar resistor yang berada pada suatu PCB. Data sample digunakan sebagai data training saat melakukan pelatihan dengan metode *template matching* dan model warna HSV.

2.2.2 Citra Masukan

Data masukan yang digunakan dalam aplikasi ini, yaitu citra resistor dengan format gambar .jpg. Selain itu warna citra juga tidak gelap agar aplikasi dapat lebih maksimal melakukan pengenalan resistor.



Gambar 2. Contoh Citra Masukan

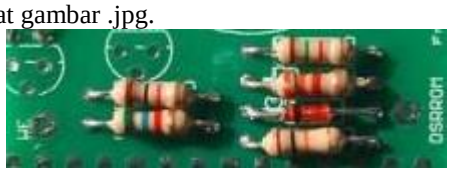
Gambar 2. merupakan gambar data masukan sebagai data latih. Berikut adalah gambar matrik warna *Red* (R), *Green* (G), dan *Blue* (B) dari gambar 2.

R=85 G=120 B=87	R=129 G=164 B=131	R=76 G=92 B=66
R=147 G=182 B=149	R=137 G=172 B=139	R=57 G=73 B=47
R=81 G=120 B=89	R=67 G=106 B=75	R=21 G=39 B=17

Gambar 3. Matriks RGB Citra Masukan

2.2.3 Citra Latih

Citra latih yang digunakan dalam aplikasi ini, yaitu citra resistor yang berada pada PCB dengan format gambar .jpg.



Gambar 4. Contoh Citra Latih

Gambar 4. merupakan gambar data masukan sebagai data latih. Berikut adalah gambar matrik warna *Red* (R), *Green* (G), dan *Blue* (B) dari gambar 4.

2.2.4 Pemodelan Warna HSV

Pada proses ini, dilakukan perubahan model warna RGB ke model warna HSV untuk citra masukan dan citra latih**.** Dengan menggunakan persamaan (2.3), (2.4), (2.5), (2.6), (2.7), (2.8), maka hasil pemodelan warna HSV untuk citra masukan adalah sebagai berikut

H=123 S=0.331 V=0.498	H=123 S=0.213 V=0.643	H=97 S=0.283 V=0.361
H=123 S=0.192 V=0.714	H=190 S=0.235 V=0.702	H=97 S=0.356 V=0.286
H=132 S=0.325 V=0.471	H=132 S=0.368 V=0.416	H=109 S=0.564 V=0.153

Gambar 5. HSV Citra Masukan

2.2.5 Kuantisasi Warna HSV

Pada ruang HSV, setiap komponen memiliki rentang nilai yang besar. Jika menggunakan nilai HSV secara langsung. Jadi lebih baik jika model warna HSV dikuantisasi secara *non-equal intervals*, yaitu H: 8 bagian, S: 3 bagian dan V: 3 bagian. Dengan menggunakan persamaan (2.9), (2.10), (2.11) maka hasil kuantisasi warna HSV untuk citra masukan sebagai berikut

H=3 S=1 V=1	H=3 S=1 V=1	H=3 S=1 V=1
H=3 S=0 V=2	H=4 S=1 V=2	H=3 S=1 V=1
H=3 S=1 V=1	H=3 S=1 V=1	H=3 S=1 V=0

Gambar 6. Kuantisasi Warna HSV

Setelah mendapatkan nilai kuatisasi, kemudian digabungkan menjadi 8x3x3 histogram dan mendapatkan vektor 72 dimensi. Kombinasi nilai h, s, v akan dinormalisasi menjadi 1 vektor. Dengan persamaan (2.12) maka hasil kuantisasi 1 vektor sebagai berikut

31	31	31
29	41	31
31	31	30

Gambar 7. Kuantisasi 1 vektor

2.2.6 *Normalized Cross-correlation*

Dengan metode *normalized cross correlation* perlu melakukan normalisasi pada kedua citra terlebih dahulu. Teknik normalisasi yang digunakan sesuai dengan tahapan normalisasi pada *normalized cross-correlation*.

Berikut adalah langkah-langkah untuk *normalized cross-correlation*:

1. Hitung rata-rata kedua citra dengan persamaan (2.1).
2. Hitung standar deviasi kedua citra dengan persamaan (2.2).

0	0	0
-1	3	0
0	0	-1

Gambar 8. Martiks Citra Template

2.2.7 Konvolusi

Konvolusi adalah sebuah proses dimana citra dimanipulasi dengan menggunakan eksternal mask (*template*) untuk menghasilkan citra yang baru. Langkah-langkahnya yaitu menempatkan citra template pada *pixel* (0,0) dan digeser hingga akhir dari *pixel citra* dengan persamaan (2.13).

2.2.8 *Matched Object*

Matched object merupakan hasil atau output dari proses konvolusi yang menghasilkan nilai maksimum dari pixel yang baru sebagai nilai yang paling mendekati dengan citra template yang berada pada citra target.

2.2.9 Perhitungan Nilai Hambatan

Pada perhitungan nilai hambatan suatu resistor ini menggunakan pemodelan warna HSV. Pemodelan warna HSV berguna sebagai batas nilai warna yang digunakan dalam mendeteksi warna dari setiap cincin yang yang ada pada resistor. Berikut batas nilai HSV yang digunakan sebagai acuan

Tabel 1. Batas Nilai HSV dari OpenCV

Warna	Hue	Saturation	Value
Hitam	0-180	0-25	0-50
Coklat	0-15	90-250	100-150
Merah	11-15 atau 171-180	176-255 atau 65-250	161-255 atau 50-150
Jingga	4-9	100-250	100-150
Kuning	20-30	130-250	100-160
Hijau	45-72	50-250	60-150
Biru	80-106	50-250	50-150
Ungu	130-155	40-250	50-150
Abu-Abu	0-180	0-50	50-80
Putih	0-180	0-15	90-140

Berdasarkan tabel 1. range nilai Hue, Saturation, Value masing-masing adalah 0-180º, 0-255 dan 255. Maka perlu adanya perubahan nilai range dari masing-masing komponen. Range masing-masing masing komponen akan diubah menjadi 0-360º,0-1, dan 0-1.

Tabel 2. Batas Nilai HSV

Warna	Hue	Saturation	Value
Hitam	0-360	0-0.1	0-0.2
Coklat	0-30	0.35-0.98	0.39-0.58
Merah	11-30 atau 342-360	0.69-1 atau 0.25-0.98	0.63-1 atau 0.2-0.58
Jingga	8-18	0.39-0.98	0.39-0.58
Kuning	40-60	0.5-0.98	0.39-0.62
Hijau	90-144	0.2-0.98	0.23-0.58
Biru	160-212	0.2-0.98	0.2-0.58
Ungu	260-310	0.15-0.98	0.2-0.58
Abu-Abu	0-360	0-0.2	0.2-0.31
Putih	0-360	0-0.05	0.35-0.54

Berikut ini adalah gambar resistor yang akan dihitung nilai hambatannya:



Gambar 9. Contoh Resistor

Pada gambar 9. nilai pixel yang diambil adalah nilai HSV. Berikut ini matriks nilai HSV pada gambar 9.

x/y	0	…	3	4	…	11
0	120 0.197 0.616	…	49 0.192 0.918	40 0.221 0.957	…	94 0.583 0.235
…	…	…	…	…	…	…
6	35 0.598 0.38	…	14 0.824 0.58	15 0.862 0.541	…	25 0.979 0.184
…	…	…	…	…	…	…
12	135 0.351 0.761	…	103 0.396 0.545	100 0.422 0.529	…	110 0.5 0.235
…	…	…	…	…	…	…
19	25 0.312 0.741	…	11 0.67 0.725	12 0.709 0.714	…	23 0.659 0.345
…	…	…	…	…	…	…
30	62 0.862 0.569	…	26 0.372 0.718	27 0.402 0.682	…	39 0.508 0.239

Gambar 10. Matriks Nilai HSV

Pada pixel (6,3) menunjukan nilai *Hue, Saturation, Value* masing-masing adalah 14, 0.824, dan 0.58 maka menunjukan warna coklat pada tabel 3.20. Begitu juga nilai HSV pada pixel (12,3) yang bernilai 108, 0.383, 0.553 yang menunjukan warna hijau. Dan pixel (19,4) yang bernilai 12, 0.709, 0,714 yang menunjukan warna merah.

Cincin pertama, kedua, dan ketiga dideklarasikan masing-masing sebagai A, B, C. Berdasarkan tabel 2. jika cincin pertama coklat maka akan diberikan nilai 1, cincin kedua berwarna hijau maka diberi nilai 5, dan cincin ketiga berwarna merah maka diberi nilai 10^2. Maka nilai hambatan pada gambar 9 bernilai 1,5 K Ohm.

2.3 Implementasi

2.3.1 Implementasi Perangkat Keras

Implementasi perangkat keras menjelaskan perangkat keras yang digunakan.

Tabel 3. Implementasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Processor	AMD E1-2500 APU
2	Memory	6 GB DDR 3
3	VGA	AMD Radeon HD 8210 Graphics Card
4	Hardisk	500 GB
5	Layar Monitor	Resolusi 1366x768 *pixel*

2.3.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk perhitungan nilai hambatan resistor pada citra digital dapat dilihat pada tabel 4.

Tabel 4. Implementasi Perangkat Lunak

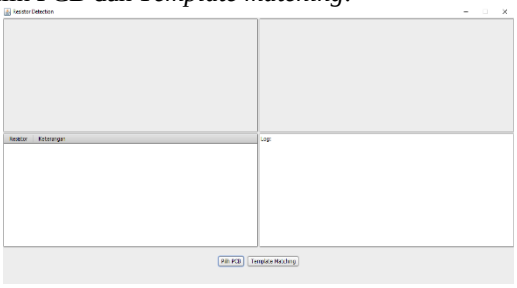
No	Perangkat Keras	Spesifikasi
1	Sistem Operasi	Windows 10 Pro 64-bit
2	Bahasa Pemrograman	Java
3	Perangkat Pendukung	Java Development Kit
4	Aplikasi Tools	Netbeans IDE 7.4
5	Permodelan	Microsoft Visio 2013

2.3.3 Implementasi Antarmuka

Implementasi antarmuka untuk menghitung nilai hambatan suatu resistor pada citra digital memiliki beberapa antarmuka. Berikut adalah antarmuka yang ada pada aplikasi.

1. Tampilan Menu Utama

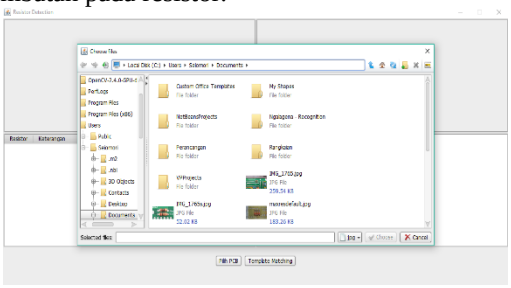
Tampilan yang pertama kali akan muncul saat program dijalankan yaitu berupa menu utama yang terdiri dari 2 tombol yaitu tombol pilih PCB dan *Template matching*.



Gambar 11. Tampilan Menu Utama

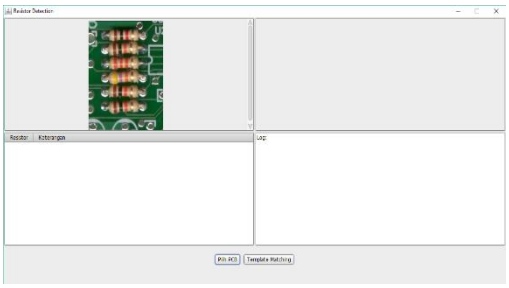
2. Tampilan Menu Input Citra

Menu input digunakan untuk memulai proses pendeteksian dan perhitungan nilai hambatan pada resistor.



Gambar 12. Tampilan Menu input Citra

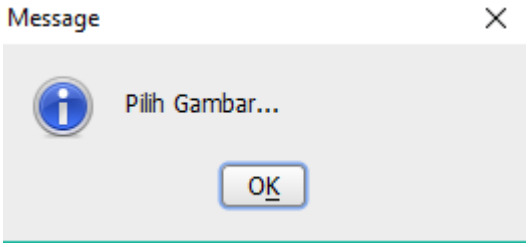
Citra yang di inputkan file berformat jpg. Tampilan menu utama setelah citra di inputkan dapat dilihat pada gambar 13.



Gambar 13. Tampilan Menu Utama

3. Tampilan Pesan Belum Memilih Citra

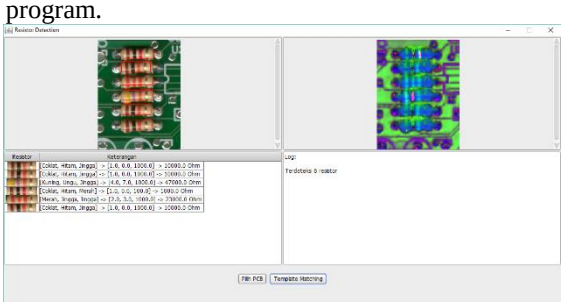
Tampilan pesan ini muncul ketika pengguna belum memilih citra. Pada gambar 14 merupakan tampilan pesan belum memilih citra.



Gambar 14. Tampilan Pesan

4. Tampilan *Output*

Tampilan *output* merupakan hasil akhir dari pendeteksian dan perhitungan nilai resistor. Pada gambar 4.5 merupakan tampilan hasil output dari program.

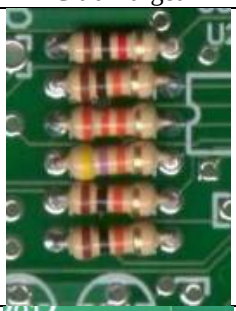


Gambar 15. Tampilan *Output*

2.3.4 Pengujian Program

Pengujian ini dilakukan pada file citra berformat jpg. Citra yang diuji sebagai target dapat dilihat pada tabel 5.

Tabel 5. Citra Uji

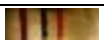
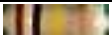
No	Citra Target	Resolusi
1		177x235
2		250x134

Tabel 5. Citra Uji

No	Citra Target	Resolusi
3		105x103
4		247x93
5		215x145

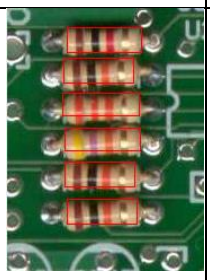
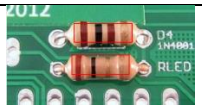
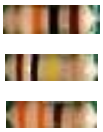
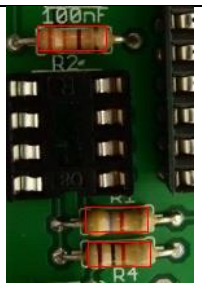
Citra template yang akan di cocokan pada target berformat jpg. Pada tabel 6 merupakan beberapa citra template.

Tabel 6. Citra Template

No	Citra Template	Resolusi	Nilai Hambatan
1		51x22	12k Ω ± 5%
2		66x22	1k Ω ± 5%
3		62x22	10k Ω ± 5%
4		68x20	23k Ω ± 5%
5		67x20	47k Ω ± 5%
6		17x51	330 Ω ± 5%
7		15x47	10k Ω ± 5%
8		16x52	47k Ω ± 5%
9		11x40	100k Ω ± 5%
10		11x40	330 Ω ± 5%
11		12x41	10k Ω ± 5%
12		77x25	100 Ω ± 5%
13		76x26	3 Ω ± 5%

Berikut merupakan hasil pengujian program yang telah dilakukan dengan menggunakan parameter pengujian seperti menghitung akurasi pendeteksian dan perhitungan nilai hambatan. Pada tabel 7. merupakan hasil pendeteksian resistor.

Tabel 7. Hasil Deteksi Citra

No	Citra Target	Citra Template	Keterangan
1			Terdeteksi
2			Terdeteksi
3			Terdeteksi
4			Terdeteksi
5			Terdeteksi

Dari hasil pengujian pendeteksian resistor dapat diperoleh kesimpulan sebagai berikut:

a. Banyak pengujian: 5
b. Jumlah resistor keseluruhan: 19
c. Jumlah citra terdeteksi: 17
d. Persentase keakurasian: $\frac{17}{19} x\ 100\% = 89\%$

Pada tabel 8 merupakan hasil perhitungan nilai hambatan dari setiap resistor

Tabel 8. Hasil Perhitungan Nilai Hambatan

No	Citra Template	R 1	R 2	R 3	R 4	Hasil Deteksi	Kesimpulan
1		✓	✓	✓	✓	1k Ω ± 5%	Benar
2		✓	✓	✓	✓	10k Ω ± 5%	Benar
3		✓	✓	✓	✓	23k Ω ± 5%	Benar
4		✓	✓	✓	-	47k Ω	Ring ke 4 tidak terdeteksi
5		✓	✓	✓	✓	200 Ω ± 5%	Benar
6		✓	✓	✓	-	3Ω	Ring ke 4 tidak terdeteksi
7		✓	✓	✓	✓	12k Ω ± 5%	Benar
8		✓	✓	✓	✓	100k Ω ± 5%	Benar
9		✓	✓	✓	✓	330 Ω ± 5%	Benar
10		-	✓	✓	✓	300 Ω	Salah
11		✓	✓	✓	✓	330 Ω	Ring ke 4 tidak terdeteksi
12		✓	✓	✓	✓	10k Ω ± 5%	Benar
13		✓	✓	✓	✓	47k Ω ± 5%	Benar

Dari hasil pengujian perhitungan nilai hambatan resistor dapat diperoleh kesimpulan sebagai berikut:

a. Banyak pengujian: 5
b. Jumlah resistor: 13
c. Jumlah resitor yang benar: 9
d. Jumlah resitor yang salah: 1
e. Jumlah resistor yang cincin ke 4 tidak terdeteksi: 3
f. Persentase keakurasian: 69%

3. PENUTUP

3.1 Kesimpulan

Berdasarkan hasil analisis, perancangan, dan implementasi maka dapat mengambil kesimpulan bahwa dari hasil pengujian, *template matching* dapat mendeteksi objek pada citra digital yang menghasilkan akurasi sebesar 89% dan untuk perhitungan nilai hambatan pada resistor sebesar 69%.

3.2 Saran

Aplikasi pendeteksi object dan perhitungan nilai resistor masih banyak kekurangan, sehingga perlu dapat dikembangkan lagi. Untuk meningkatkan lagi kualitas akurasi dan fungsionalitas dari aplikasi ini, ada beberapa saran yang perlu dipertimbangkan diantaranya:

1. Pendeteksian akan lebih baik lagi jika dalam mendeteksi objek menggunakan resolusi citra yang lebih tinggi dan kualitas citra dengan format citra yang lain.
2. Aplikasi akan lebih baik jika dapat mendeteksi objek dengan model template yang berbeda seperti rotasi template yang berbeda dengan citra target.
3. Pendeteksian objek memiliki kekurangan yaitu nilai pixel template harus menyerupai nilai pixel yang ada pada target agar objek dapat terdeteksi dengan benar.
4. Perhitungan resistor akan lebih baik lagi jika range warna yang terdeteksi diinisialisasi dengan benar.

DAFTAR PUSTAKA

[1] Muda, Imam N. 2013. Elektronika Dasar. Malang : Gunung Samudera.

[2] Putu, I Pratama Andika, dkk. 2015. Rancang Bangun Aplikasi Pendeteksi Tipe Dan Nilai Resistor Berbasis Android. Bali : Lontar Komputer Vol.6, No.1.

[3] Munir, Rinaldi. 2004. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: Informatika.

[4] Pressman, Roger S., & Bruce R. Maxim. 2014. *Software Engineering a Practitioner's Approach.*

[5] Gonzales, R.C., & Woods, R.E. 2001. *Digital Image Processing, Second Edition*. United State of American: Prentice Hall.

[6] Ferderico, Muhammad S. 2017. Implementasi Metode *Support Vector Machine (SVM)* dan *Zoning* pada pengolahan citra untuk pengenalan citra tulisan tangan. Skripsi. Jurusan Teknik Informatika. Universitas Komputer Indonesia.

[7] Putra, Erwin Dwika. 2012. Algoritma *Template matching* Pada *OCR (Optical Character Recognition).* Skripsi. Jurusan Teknik Informatika. Universitas Muhammadiyah Bengkulu.

[8] Setiawan, Heri. 2014. Implementasi Metode *Normalized Cross Correlation* Pada *Template matching* Untuk Aplikasi Pendeteksi Objek Pada Citra Digital. Skripsi. Jurusan Teknik Informatika. Universitas Komputer Indonesia.

[9] R. A. a. M.Shalahudin, 2014. Rekayasa Perangkat Lunak 'Terstruktur dan Berorientasi objek', Bandung: Informatika.

[10] R. Sianipar, 2013. Teori dan Implementasi Java, Bandung: Informatika.