

IMPLEMENTASI PENGOLAHAN CITRA UNTUK MENGHITUNG RESISTANSI RESISTOR MENGGUNAKAN METODE BACK PROPAGATION

Abdi Haqqi An Nazilli¹, Deddy Kusbianto Purwoko Aji², Ulla Delfana Rosiani³

^{1,2}Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang

¹abdiqh@hotmail.com, ²deddy_kusbianto@polinema.ac.id, ³rosi@polinema.ac.id

Abstrak

Dalam perhitungan nilai resistansi pada resistor, dibutuhkan pembacaan warna sesuai gelang yang tertera, ketepatan membaca warna gelang pada resistor sangat berpengaruh. Setelah itu, juga dibutuhkan secara otomatis penghitungan nilai warna pada setiap gelang sehingga akan menghasilkan nilai resistansi dan toleransi resistor. Namun dengan berkembangnya teknologi yang semakin maju maka dimungkinkan untuk menghitung resistansi resistor dengan bantuan sebuah komputer.

Resistansi resistor dapat dihitung melalui ekstraksi fitur RGB dari masing-masing warna gelang resistor. Setelah itu nilai tersebut akan digunakan sebagai inputan dalam proses pembelajaran dengan menggunakan perhitungan back-propagation agar menghasilkan bobot terbaik. Bobot terbaik itulah yang digunakan sebagai bobot dalam perhitungan proses klasifikasi. Sehingga akan didapatkan keluaran yakni 11 warna resistor. Tingkat keakuratan klasifikasi warna gelang menggunakan back-propagation ini yaitu 40% dan Tingkat keakuratan penghitungan resistansi resistor ini yaitu 10%.. Namun perlu diperhatikan tingkat keberhasilan sangat dipengaruhi oleh faktor variasi data masukan pada warna gelang resistor yang diklasifikasikan.

Kata Kunci : Resistor, Jaringan Syaraf Tiruan, Back-propagation

1. Pendahuluan

Resistor adalah salah satu komponen elektronika yang berfungsi sebagai penahan arus yang mengalir dalam suatu rangkaian dan memiliki karakteristik utama yaitu resistansi dan toleransi, yang dapat dihitung melalui cincin warna yang ada pada resistor tersebut.

Dalam perhitungan nilai resistansi pada resistor, dibutuhkan pembacaan warna sesuai gelang yang tertera, ketepatan membaca warna gelang pada resistor sangat berpengaruh. Setelah itu, juga dibutuhkan secara otomatis penghitungan nilai warna pada setiap gelang sehingga akan menghasilkan nilai resistansi dan toleransi resistor.

Maka sesuai perkembangan teknologi saat ini pemanfaatan teknologi harus dilakukan dengan maksimal untuk membantu masalah ini. Dalam hal ini khususnya adalah Pengolahan Citra Digital yang dapat dimanfaatkan untuk membaca warna – warna pada gelang resistor. Dan juga menghitung resistansi yang sudah ditetapkan dengan hasil yang diharapkan.

Penulis akan membuat sistem untuk menghitung resistansi resistor. Dalam hal ini penulis menggabungkan sebuah kecerdasan buatan dengan pengolahan citra, agar dapat menirukan kerja mata dan otak kita yang saling bekerja sama

untuk mengenali obyek, yaitu Jaringan Syaraf Tiruan (JST). Dalam penelitian kali ini, penulis menggunakan metode back-propagation. Pengolahan citra akan bertugas membaca warna pada setiap gelang dan metode back-propagation akan melatih jaringan sesuai pola masukan. Selama pelatihan, jaringan akan dilatih untuk mengenali pola warna – warna gelang, serta jaringan ini mampu memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan.

2. Landasan Teori

2.1. Resistor

Resistor disebut juga dengan tahanan atau hambatan, berfungsi untuk menghambat arus listrik yang melewatinya. Satuan resistor adalah Ohm. ($1 \text{ M}\Omega$ (mega ohm) = 1000 K Ω (kilo ohm) = 106 Ω (ohm)). Resistor terbagi menjadi dua macam, yaitu : 1. Resistor tetap, 2. Resistor variable. Resistor tetap yaitu resistor yang nilai hambatannya relatif tetap, biasanya terbuat dari karbon, kawat atau paduan logam.

2.2. Pengolahan Citra

Pengolahan citra adalah suatu pemrosesan citra dengan bantuan sebuah computer sehingga menjadikan kualitas citra tersebut menjadi lebih baik. Dengan kata lain

pengolahan citra merupakan kegiatan memperbaiki kualitas citra agar mudah diinterpretasikan oleh manusia atau computer.

2.3. Warna RGB

Setiap *pixel* 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 tingkatan warna. Sehingga setiap *pixel* mempunyai kombinasi warna sebanyak 16 juta lebih. Itulah sebabnya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna alam.

2.4. Sampling

Sampling digunakan untuk memilih dan mengisolasi suatu bagian dari suatu citra. Dimana proses ini akan menurunkan jumlah *pixel* dalam citra tersebut dan menghilangkan sebagian informasi dari citra tersebut. Hal ini akan membuat ukuran citra berubah menjadi lebih kecil dan akan lebih memudahkan dalam perhitungan nilai warna yakni RGB.

2.5. Metode Back-Propagation

Back-Propagation merupakan bagian dari jaringan syaraf tiruan dengan cara melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan, serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa (tapi tidak sama) dengan pola yang dipakai selama pelatihan. (Jong Jek Siang:2009)

3. Analisis dan Perancangan

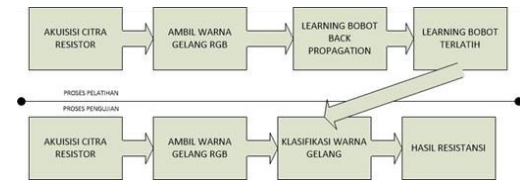
Dalam analisis dan perancangan akan membahas tentang seperti apa analisa dan perancangan dari aplikasi ini

3.1. Analisis Sistem

Analisis sistem merupakan sebuah analisa penjabaran tentang komponen yang dibutuhkan oleh sistem, baik itu perangkat lunak maupun perangkat keras. Serta membahas tentang gambaran umum dari sistem yang akan dibuat.

3.2. Gambaran Umum Aplikasi

Aplikasi yang dibangun yakni pengolahan citra untuk menghitung resistansi resistor metode back-propagation.



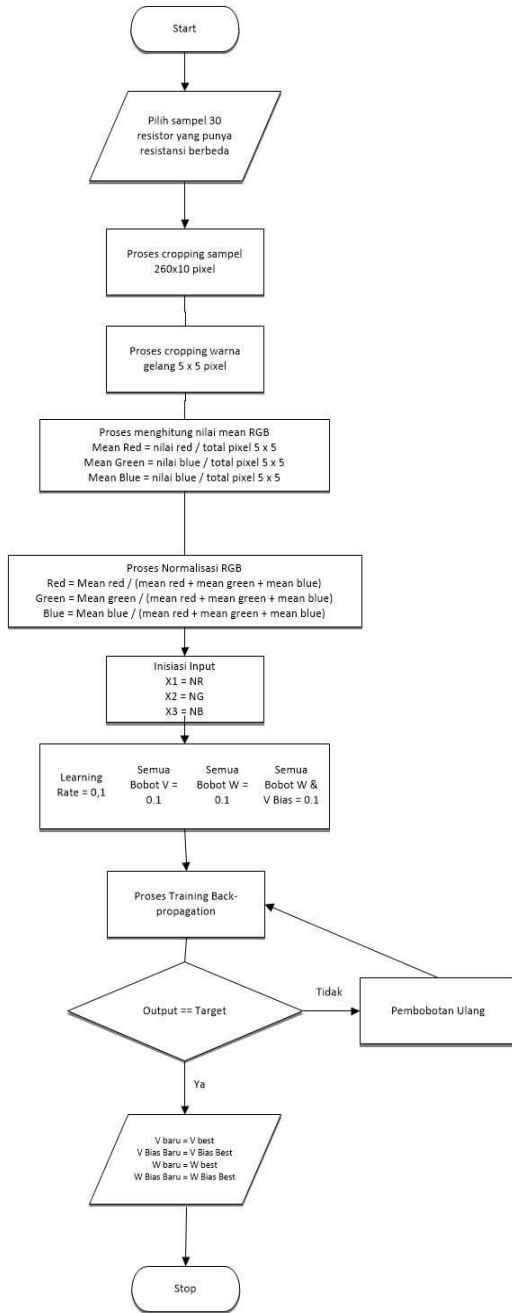
Gambar 3.1 Gambaran Umum Aplikasi

3.3. Perancangan Sistem

Perancangan sistem merupakan suatu proses desain sistem dalam penggambaran dan pembuatan sketsa *interface* aplikasi hingga perhitungan dari metode itu sendiri yakni *back-propagation*. Rancangan ini sendiri akan terbagi menjadi tiga yakni perancangan perhitungan metode, perancangan proses dalam bentuk *flowchart*, dan perancangan *user interface* atau *mockup* dari aplikasi ini.

3.4. Perancangan Proses Training

Dalam proses *training* dapat dijabarkan oleh gambar flowchart di bawah ini, dari proses membuka citra resistor hingga mendapatkan bobot.



Gambar 3.2 Proses Training

3.5. Perancangan Proses Testing

Dalam proses *testing* mulai dari sample resistor dimasukkan lalu dilakukan proses cropping 260x10 pixel dan dilakukan cropping lagi 5x5 pixel maka akan menghasilkan nilai RGB warna pergelang.

Setelah itu memasukkan bobot yang telah dilatih sehingga masing – masing nilai warna RGB akan diklasifikasikan sesuai hasil perhitungan metode. Nilai yang telah dihitung akan diklasifikasi sebanyak 11 warna. Jika warna sudah didapatkan maka penghitungan resistansi akan secara otomatis didapatkan.

4. Implementasi

4.1. Implementasi Open Image

Proses open image merupakan langkah awal dalam memulai menghitung resistansi resistor. Dikarenakan pada citra resistor tersebut akan dilakukan proses ekstraksi komponen *Red*, *Green*, dan *Blue* (RGB) yang ada di dalamnya.

4.2. Implementasi Cropping Image

Pada proses cropping ini citra akan dilakukan proses cropping sebesar 260x10 pixel dan 5x5 pixel.

4.3. RGB

Untuk menghitung nilai normalisasi dari RGB citra hasil cropping dapat dipakai persamaan di bawah ini :

$$NR = \frac{MR}{MR+MG+MB} \quad (4.1)$$

Keterangan :

NR = Normalisasi Red

MR = Mean Red

MG = Mean Green

MB = Mean Blue

4.4. Implementasi Metode Back-Propagation

Pada sub bab ini menjelaskan bagaimana penulis melakukan implementasi perhitungan training serta testing.

4.4.1. Implementasi Training

Berikut merupakan langkah-langkah yang dilakukan penulis untuk melakukan perhitungan training :

1. Setelah citra berhasil dilakukan proses cropping sebesar 260x10 pixel dan diketahui nilai normalisasi RGB dari citra tersebut maka lakukan langkah 2.
2. Beri bobot acak kecil untuk tiap neuron, bias, learning rate (dalam hal ini penulis menggunakan algoritma dari Nguyen-Widrow untuk menentukan bobot acak kecil).
3. Menghitung semua keluaran pada node hidden layer

$$Y_{net0} = BiasV00 + (NR * V00) + (NG * V10) + (NB * V20) \quad (4.2)$$

$$Y_0 = \frac{1}{(1+EXP^{-Y_{net0}})} \quad (4.3)$$

4. Menghitung keluaran pada output layer

$$Z_{net} = BiasW00 + (Y1 * W00) + (Y2 * W10) + (Y3 * W20) \quad (4.4)$$

$$Z = \frac{1}{(1+EXP^{-Z_{net}})} \quad (4.5)$$

5. Hitung factor kesalahan dari Z

$$\delta_Z = Z * (T - Z) * (1 - Z) \quad (4.6)$$

6. Hitung suku perubahan bobot pada neuron dari hidden layer ke output

$$\Delta BiasW00 = \alpha * \delta_Z * BiasW00 \quad (4.7)$$

$$\Delta W00 = \alpha * \delta_Z * Y0$$

7. Menghitung faktor kesalahan hidden layer

$$\delta_{netY0} = \delta_Z * W00 \quad (4.8)$$

$$\delta_{Y0} = \delta_{netY0} * Y0 * (1 - Y0) \quad (4.9)$$

8. Hitung suku perubahan bobot pada neuron yang menuju hidden layer

$$\Delta BiasV00 = \alpha * \delta_{Y1} * BiasV00 \quad (4.10)$$

$$\Delta V00 = \alpha * \delta_{Y0} * NR \quad (4.11)$$

9. Hitung bobot baru dari neuron

$$BiasW0 \text{ new} = BiasW0 + \Delta BiasW0 \quad (4.12)$$

$$BiasV00 \text{ new} = BiasV00 + \Delta BiasV00 \quad (4.13)$$

Proses diatas dilakukan terus menerus hingga menemukan epoch yang konvergen dengan target dari data.

4.4.2. Implementasi Testing

Setelah bobot hasil training didapatkan maka pada saat testing akan dilakukan satu kali perhitungan dengan menggunakan metode back-propagation dan hasil dari perhitungan itu akan diklasifikasikan sesuai dengan ketentuan warna berikut :

1. Hitam : 1
2. Coklat : 0.9091
3. Merah : 0.8182
4. Orange : 0.7273
5. Kuning : 0.63
6. Hijau : 0.5455
7. Biru : 0.4546
8. Ungu : 0.3637
9. Abu – Abu : 0.2728
10. Putih : 0.1819
11. Emas : 0.091

5. Uji Coba dan Pembahasan

Pada bab pengujian dan pembahasan ini akan dilakukan tahapan untuk menguji hasil dari implementasi sistem yang telah dilakukan.

5.1. Pengujian Sistem

Pengujian sistem ini dilakukan dengan cara menjalankan aplikasi secara detail pada setiap menu yang ada, dengan tujuan untuk mengetahui menu atau fitur mana yang sudah berfungsi dengan baik maupun yang tidak berfungsi sesuai dengan sebagaimana mestinya.

5.2. Pengujian Training

Dari pengujian training didapatkan bobot yang mendekati dengan konvergensi jaringan dengan banyak epoch yakni 10 juta epoch. Di bawah ini merupakan gambar dari epoch ke 10 juta. Serta bobot yang didapatkan.

	1	0.72308027...	0.66357729...	0.16235536...	6.31268722...
1000000	1	0.99991129...	0.87282143...	0.90858537...	-0.0067159...
	1	0.99654522...	0.83197318...	0.95599012...	-0.0036014...
	1	0.99996429...	0.87495400...	0.90731479...	0.00786068...
	1	0.99999999...	0.92568892...	0.76601427...	-0.0017338...
	1	0.99999221...	0.88829507...	0.87896717...	0.00540677...
	1	0.99999999...	0.95944314...	0.61314126...	0.00041346...
	1	0.99999999...	0.97663352...	0.52376585...	-0.0016920...
	1	0.03699059...	0.65028398...	2.51538771...	2.51538770...
	1	0.72308035...	0.66357739...	0.16235536...	6.31268747...

Gambar 5.1 Epoch ke 1 Juta

	V11	V12
	1.5653175720297	-0.76627335122...

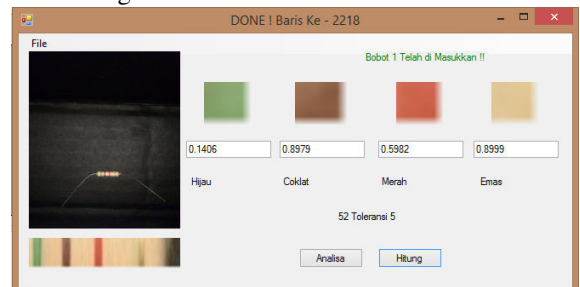
Gambar 5.2 Bobot Final Neuron V

Bobot Hidden Layer ke Output Layer				
	W1	W2	W3	Wbias
	9.446157...	-30.2353...	21.18425...	4.905235...

Gambar 5.3 Bobot Final Neuron W

5.3. Pengujian Testing

Setelah dilakukan training dan mendapatkan bobot. Maka selanjutnya dilakukan pengujian testing dari aplikasi. Di bawah ini merupakan hasil dari uji coba testing.



Gambar 5.4 Hasil Testing pada Reistor

5.4. Pengujian Akurasi Klasifikasi Warna dan Penghitungan Resistor

Dengan *prosentase* keakuratan secara keseluruhan dapat dihitung menggunakan rumus berikut:

$$\frac{\text{Jumlah data terklasifikasi}}{\text{Jumlah pengambilan seluruh data}} \times 100\%$$

1. Maka didapatkan tingkat akurasi warna sebagai berikut :

$$\frac{12}{30} \times 100\% = 40\%$$

2. Maka didapatkan tingkat akurasi penghitungan resistansi sebagai berikut :

$$\frac{1}{10} \times 100\% = 10\%$$

Sehingga tingkat akurasi penghitungan klasifikasi warna resistor secara keseluruhan adalah sebesar 40 % dan tingkat akurasi penghitungan resistansi resistor secara keseluruhan adalah sebesar 10 %.

6. Kesimpulan

6.1. Kesimpulan

Berdasarkan dari hasil penelitian yang dilakukan oleh penulis, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Implementasi Sistem Implementasi Pengolahan Citra Untuk Menghitung Resistansi Resistor Menggunakan Metode Back-Propagation yang dibangun sesuai dengan perancangan yang telah dibuat.
2. Aplikasi masih jauh dari sempurna, dan dapat digunakan sebagai acuan pada penelitian lebih lanjut.
3. Tingkat keberhasilan klasifikasi gelang warna menggunakan metode back-propagation ini berhasil memperoleh nilai 40% dan Tingkat keberhasilan penghitungan resistansi resistor ini berhasil memperoleh nilai 10%.
4. Klasifikasi setiap warna gelang resistor mempunyai akurasi yang berbeda. Hal itu dikarenakan dari banyaknya epoch dan data training.

6.2. Saran

Saran yang diberikan untuk pengembangan sistem pada penelitian selanjutnya adalah sebagai berikut :

1. Diharapkan pada pengembangan aplikasi selanjutnya dapat menggunakan sistem realtime dalam akuisisi citra resistor.
2. Diharapkan pula pada pengembangan aplikasi selanjutnya dapat mendeteksi warna secara lengkap.

3. Diharapkan pada pengembangan aplikasi selanjutnya dapat menambah jumlah node pada hidden layer sehingga jaringan dapat lebih detail dalam melakukan pembelajaran.

Daftar Pustaka :

- Willem. 2013. *Teknik Listrik Dasar Otomotif*. Kementerian Pendidikan Dan Kebudayaan Republik Indonesia.
- Andika, I Putu P., et all., 2015, *Rancang Bangun Aplikasi Pendeteksi Tipe Dan Nilai Resistor Berbasis Android*. Universitas Udayana. VOL.6 (1), 529-540.
- Derisma, Deswari, Hendrick. “*Identifikasi Kematangan Buah Tomat Menggunakan Metode Back Propagation*”.
- Jong Jek Siang. 2009. “*Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*”. Yogyakarta : Penerbit Andi.
- Kusumaningtyas. Sella. 2015. Identifikasi Kematangan Buah Tomat Berdasarkan Warna Menggunakan Metode Jaringan Syaraf Tiruan (JST). Malang : Politeknik Negeri.
- Kusumanto, Novi. 2011. “*Pengolahan citra digital untuk mendeteksi obyek menggunakan pengolahan warna model normalisasi RGB*”. Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2011.
- Pandjaitan, L. W., 2007. “*Dasar-dasar Komputasi Cerdas*”. Andi Offset, Yogyakarta.
- Siregar, Tantry Meilany; dkk. 2015. “*Identifikasi Kematangan Buah Pisang (Musa Paradisiaca) Dengan teknik Jaringan Syaraf Tiruan*”. J.Rekayasa Pangan dan Pert., Vol.3 No. 2 Th. 2015
- Zekson. 2013. “*Jaringan syaraf tiruan dengan algoritma backpropagation untuk penentuan kelulusan sidang skripsi*”. Pelita Informatika Budi Darma, Volume : IV, Nomor: 1, Agustus 2013