

DETEKSI WARNA KULIT MENGGUNAKAN MODEL WARNA *CIELAB* NEURAL NETWORK UNTUK IDENTIFIKASI RAS MANUSIA (STUDI KASUS RAS: KAUKASOID, MONGOLOID, DAN NEGROID)

Harvini Lazi¹, Rusdi Efendi², Endina Putri Purwandari³

^{1,3}Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu
Jalan W.R Supratman Kandang Limun Bengkulu 38371A INDONESIA
(telp: 0736-341022; fax: 0736-341022)

¹harvinilazi@yahoo.co.id

³endinaputri@unib.ac.id

Abstrak: Penelitian ini bertujuan untuk melakukan deteksi warna kulit untuk mengidentifikasi ras manusia dari citra digital. Metode yang digunakan dalam penelitian ini adalah model warna *CIELAB* dan *backpropagation neural network*. Model warna *CIELAB* digunakan untuk melakukan proses deteksi warna kulit dan ekstraksi fitur warna, sedangkan metode *backpropagation neural network* digunakan untuk mengidentifikasi jenis ras dari citra masukan. Citra yang digunakan dalam penelitian ini adalah citra dari ras Kaukasoid, Mongoloid, dan Negroid. Aplikasi pengidentifikasi ras manusia ini dibangun dengan menggunakan bahasa pemrograman MatLab R2013b. Aplikasi model warna *CIELAB* dan *Backpropagation Neural Network* ini menghasilkan tingkat akurasi: (1) Pengujian optimasi jaringan pelatihan dengan menggunakan jumlah *neuron* 30 dan fungsi aktivasi *logsig* memiliki keakurasian paling baik dari pengujian optimasi lainnya dengan nilai akurasi sebesar 93,3%; (b) Pengujian dengan menggunakan 30 citra masukan yang berasal dari dataset dengan nilai akurasi sebesar 93,33 %; dan (3) Pengujian dengan menggunakan 15 citra masukan yang berasal dari foto sendiri serta pengambilan dari internet dengan nilai akurasi sebesar 93,33 %.

Kata kunci: deteksi, warna kulit, *CIELAB*, *backpropagation neural network*, ras manusia.

Abstract: The research is used to detect skin color to identify the human races from digital image. The method used in this research is CIELAB color model backpropagation neural network. CIELAB color model used to perform skin color detection process and backpropagation neural network uses to identify the type of human races. Imagery used in this study is the human image that contains the color of skin was Caucasian, Mongoloid, and Negroid. Application identifiers human race is built using a programming language MatLab R2013b. The application of CIELAB and Backpropagation Neural Network color models results in an accuracy of: (1) Testing of training network optimization using the number of 30 neurons and logsig activation function has the best accuracy from other optimization testing with an accuracy

of 93.3%; (b) *Testing using 30 input images derived from the dataset with an accuracy value of 93.33%; and (3) Tests using 15 input images derived from the photographs themselves and internet retrieval with an accuracy of 93.33%.*

Keywords: *detection, skin color, CIELAB, backpropagation neural network, human races*

I. PENDAHULUAN

Perkembangan teknologi pengolahan citra digital memberikan kemudahan untuk memproses suatu citra agar dapat dimanfaatkan dalam berbagai bidang. Salah satu pengolahan dalam bidang citra digital adalah deteksi kulit atau segmentasi kulit. Kulit dapat digunakan sebagai sumber informasi untuk proses pendeteksian yaitu pendeteksian manusia, pendeteksian wajah, deteksi gerakan tangan, dan lain-lain.

Kulit dapat dijadikan sebagai sumber awal informasi yang baik karena memungkinkan pemrosesan yang cepat dan langsung, tahan terhadap variasi geometri dari pola-pola/tekstur kulit, tahan terhadap perubahan resolusi citra dan mengurangi ketergantungan pada penggunaan peralatan penjejak yang khusus. Selain itu kulit juga memiliki karakteristik warna, hal ini memberikan kemudahan karena warna merupakan salah satu aspek yang dapat dengan mudah dan cepat untuk dikenali manusia [1].

II. LANDASAN TEORI

A. Ras Manusia

Penyebaran populasi manusia yang terdapat di berbagai macam benua di dunia menyebabkan adanya variasi manusia yang membedakan antara manusia satu dan lainnya. Perbedaan atau variasi dari populasi manusia ini diakibatkan dari perbedaan frekuensi sifat biologis yang mereka miliki, baik itu yang dapat dilihat ataupun yang tak

terlihat. Karakter eksternal atau yang dapat dilihat antara lain adalah warna kulit, tinggi badan, dan bentuk tubuh yang mereka miliki. Sedangkan karakter internal biologis yang tidak dapat dilihat adalah seperti kerentanan terhadap penyakit [2].

Ras merupakan suatu konsep yang penting untuk memudahkan pemikiran dalam mempelajari variasi populasi manusia ini. Meskipun terdapat beberapa karakter eksternal yang membedakan antara ras satu dengan yang lainnya, namun perbedaan yang paling menonjol dan dapat dilihat dengan mudah adalah adanya perbedaan warna kulit antara ras-ras yang berbeda. Secara tradisional ras manusia oleh para pakar dibedakan atas tiga ras utama [2], yaitu sebagai berikut:

1) Ras Kaukasoid

Ras kaukasoid atau berkulit putih, adalah ras manusia yang sebagian besar menetap di Eropa, Afrika Utara, Timur Tengah, Pakistan, dan India. Secara umum ras ini memiliki ciri fisik [3], sebagai berikut:

- a) berkulit putih
- b) tekstur bibir tipis
- c) memiliki bulu yang tebal
- d) rambut lurus atau bergelombang

Beberapa contoh citra ras Kaukasoid dari dataset SFA (*Skin Face Analysis*) ditunjukkan pada Gambar 2.1 berikut ini.



Gambar 2.1 Citra dari Ras Kaukasoid

2) Ras Mongloid

Ras Mongloid atau berkulit kuning, adalah ras manusia yang sebagian besar menetap di Asia Utara, Asia Timur, Asia Tenggara, Madagaskar di lepas pantai timur Afrika,

beberapa bagian India Timur Laut, Eropa Utara, Amerika Utara, Amerika Selatan, dan Oseania. Secara umum ras ini memiliki ciri fisik [3], sebagai berikut:

- kulit kuning
- kelopak mata terdapat *plica marginalis*
- mata berwarna coklat sampai hitam
- rambut berwarna hitam dan lurus
- dahi kecil dan tegak.

Beberapa contoh citra ras Mongoloid dari dataset SFA (*Skin Face Analysis*) ditunjukkan pada Gambar 2.2 berikut ini.



Gambar 2.2 Citra dari ras Mongoloid

3) Ras Negroid

Ras Negroid atau berkulit hitam, adalah ras manusia yang terutama mendiami benua Afrika di sebelah gurun Sahara. Secara umum ras ini memiliki ciri-ciri fisik [3], sebagai berikut:

- pigmentasi kulit yang kuat (kulit hitam)
- bibir dan hidung lebar dan tebal
- rambut keriting
- mata berwarna coklat sampai hitam.

Beberapa contoh citra ras Negroid dari dataset SFA (*Skin Face Analysis*) ditunjukkan pada Gambar 2.3 berikut ini.



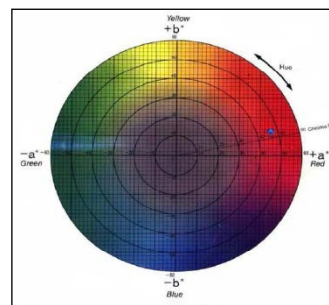
Gambar 2.3 Citra dari Ras Negroid

B. Model Warna CIELAB ($CIE L^*a^*b^*$)

Ruang warna CIE $L^*a^*b^*$ atau dikenal dengan nama CIELAB adalah ruang warna yang paling lengkap yang ditetapkan oleh Komisi Internasional

tentang iluminasi warna (*Commission Internationale de leclairage*, dikenal sebagai CIE). Ruang warna ini mampu menggambarkan semua warna yang dapat dilihat dengan mata manusia dan seringkali digunakan sebagai referensi ruang warna [4].

CIELAB merupakan model warna yang dirancang untuk menyerupai persepsi penglihatan manusia. Diagram warna CIELAB ditunjukkan pada gambar 2.4 di bawah ini:



Gambar 2.4 Diagram Warna CIELAB [5]

Pada model warna CIELAB terdapat tiga komponen warna yaitu L sebagai *luminance* (pencahayaan), a dan b sebagai dimensi warna yang berlawanan. Diagram warna CIELAB pada gambar 2.4 di atas dapat diketahui lokasi warna pada sistem ditentukan dengan koordinat L^* , a^* , dan b^* . Notasi L^* : 0 (hitam); 100 (putih) menyatakan cahaya pantul yang menghasilkan warna akromatik putih, abu-abu, dan hitam. Notasi a^* : warna kromatik campuran merah-hijau dengan nilai $+a^*$ (positif) dari 0 sampai $+80$ untuk warna merah dan nilai $-a^*$ (negatif) dari 0 sampai -80 untuk warna hijau. Notasi b^* : warna kromatik campuran biru-kuning dengan nilai $+b^*$ (positif) dari 0 sampai $+70$ untuk warna kuning dan nilai $-b^*$ (negatif) dari 0 sampai -70 untuk warna biru [5].

Dalam proses konversi model warna RGB ke model warna CIELAB terlebih dahulu dilakukan proses konversi warna RGB ke XYZ, kemudian

dilakukan konversi model warna XYZ ke CIELAB.

Konversi RGB ke XYZ dimulai dengan menormalisasikan koordinat RGB dengan menggunakan persamaan (2.1) berikut ini:

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \frac{1}{255} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.1)$$

Selanjutnya, rgb didefinisikan berdasarkan persamaan (2.2) berikut ini:

$$f(n) = \begin{cases} \frac{n}{12,92} & , \text{jika } n < 0,04045 \\ \left(\frac{n+1,055}{1,055}\right)^{2,44} & , \text{untuk yang lain} \end{cases} \quad (2.2)$$

Setelah didapatkan hasil normalisasi dari komponen r, g, b, maka proses selanjutnya adalah transformasi rgb ke XYZ dengan menggunakan persamaan (2.3) berikut ini:

$$\begin{aligned} X &= 0,4124565r + 0,3575761g + 0,1804375b \\ Y &= 0,2126729r + 0,7151522g + 0,0721750b \\ Z &= 0,0193339r + 0,1191920g + 0,9503041b \end{aligned} \quad (2.3)$$

Selanjutnya dilakukan transformasi koordinat warna XYZ ke CIELAB dengan menggunakan persamaan (2.4) berikut ini :

$$\begin{bmatrix} X_n \\ Y_n \\ Z_n \end{bmatrix} = \frac{1}{100} \begin{bmatrix} 95,0470 \\ 100,000 \\ 108,880 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.4)$$

Selanjutnya, L*a*b* didefinisikan berdasarkan persamaan (2.5) berikut ini:

$$\begin{aligned} L^* &= 116 f(Y_n) - 16 \\ a^* &= 500 [f(X_n) - f(Y_n)] \\ b^* &= 200 [f(Y_n) - f(X_n)] \end{aligned} \quad (2.5)$$

Dalam hal ini, $f(q)$ dihitung berdasarkan persamaan (2.6) berikut ini:

$$f(q) = \begin{cases} q^{\frac{1}{3}}, & \text{jika } q > 0,008856 \\ 7,787q + \frac{16}{116}, & \text{untuk yang lain} \end{cases} \quad (2.6)$$

C. Backpropagation Neural Network

Neural Network atau jaringan syaraf tiruan merupakan suatu sistem pemrosesan informasi yang memiliki karakteristik yang mirip dengan jaringan syaraf biologi. Jaringan syaraf tiruan dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi [5] dengan asumsi bahwa:

- 1) Pemrosesan informasi terjadi pada banyak elemen sederhana (*neuron*).
- 2) Sinyal dikirimkan diantara *neuron-neuron* melalui perhubung-penghubung.
- 3) Penghubung antar *neuron* memiliki bobot yang akan memperkuat atau memperlemah sinyal.
- 4) Untuk menentukan output, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier). Besarnya output ini selanjutnya dibandingkan dengan suatu ambang batas.

Salah satu metode pelatihan dalam *neural network* adalah pelatihan terbimbing (*supervised learning*). Pada pelatihan terbimbing diperlukan sejumlah masukan dan target yang berfungsi untuk melatih jaringan hingga diperoleh bobot yang diinginkan. Pada setiap kali pelatihan, suatu *input* diberikan ke jaringan. Jaringan akan memproses dan mengeluarkan keluaran. Selisih antara keluaran jaringan dengan target merupakan kesalahan yang terjadi. Jaringan akan memodifikasi bobot sesuai dengan kesalahan tersebut. *Backpropagation* merupakan metode yang menggunakan *supervised learning* [6].

Algoritma pelatihan *backpropagation* adalah sebagai berikut:

- a) Inisialisasi bobot (ambil awal dengan nilai random yang cukup kecil)
- b) Tetapkan: Maksimal *Epoch*, *Target Error*, dan *Learning rate*
- c) Inisialisasi: $Epoch = 0$, $MSE = 1$.
- d) Kerjakan langkah berikut selama ($Epoch < \text{Maksimum } Epoch$) dan ($MSE > \text{Target Error}$):

1. $Epoch = Epoch + 1$

2. Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, dikerjakan:

Fase I: Propagasi maju

- a) Tiap-tiap unit masukan ($x_i = 1, 2, 3, \dots, n$) menerima signal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (layer tersembunyi).
- b) Hitung semua keluaran pada lapisan tersembunyi z_j ($j = 1, 2, 3, \dots, p$), menggunakan persamaan (2.12) di bawah ini:

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (2.12)$$

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \quad (2.13)$$

- c) Hitung semua keluaran jaringan di unit y_k ($k = 1, 2, 3, \dots, m$), menggunakan persamaan (2.14) di bawah ini:

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (2.14)$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (2.15)$$

Fase II: Propagasi mundur

- d) Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran Y_k ($k = 1, 2, 3, \dots, m$) menggunakan persamaan (2.16) di bawah ini:

$$\begin{aligned} \delta &= (t_k - y_k) f'(y_{net_k}) \\ &= (t_k - y_k) y_k (1 - y_k) \end{aligned} \quad (2.16)$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer di bawahnya (langkah e).

Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk merubah bobot w_{kj}) dengan laju percepatan α

$$\begin{aligned} \Delta w_{kj} &= \alpha \delta_k z_j; k = 1, 2, \dots, m; j \\ &= 0, 1, \dots, p \end{aligned} \quad (2.17)$$

- e) Hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi Z_j ($j = 1, 2, 3, \dots, p$) menggunakan persamaan (2.18) di bawah ini:

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{kj} \quad (2.18)$$

Faktor δ unit tersembunyi:

$$\begin{aligned} \delta_j &= \delta_{net_j} f'(z_{net_j}) \\ &= \delta_{net_j} z_j (1 - z_j) \end{aligned} \quad (2.19)$$

Hitung suku perubahan bobot v_{ji} (yang akan dipakai nanti untuk merubah bobot v_{ji}), menggunakan persamaan (2.20) di bawah ini:

$$\begin{aligned} \Delta v_{ji} &= \alpha \delta_j x_i; j = 1, 2, \dots, p; i = 0, 1, \dots, \\ &n \end{aligned} \quad (2.20)$$

Fase III: Perubahan bobot

- f) Hitung semua perubahan bobot
Perubahan bobot garis yang menuju ke unit keluaran, menggunakan persamaan (2.21) di bawah ini:

$$w_{kj}(\text{baru}) = v_{kj}(\text{lama}) + \Delta w_k \quad (2.21)$$

Perubahan bobot garis yang menuju ke unit tersembunyi, menggunakan persamaan (2.22) di bawah ini:

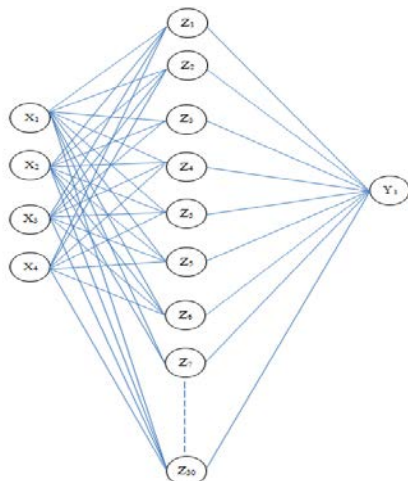
$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.22)$$

III. ANALISIS DAN PERANCANGAN

A. Arsitektur Jaringan

Arsitektur pelatihan jaringan adalah perancangan pelatihan jaringan pembelajaran yang akan dibangun menggunakan metode *backpropagation neural network* yang digunakan untuk mengenali pola tertentu, dalam hal ini jaringan pelatihan yang akan di bangun adalah jaringan pembelajaran untuk mengenali ras dari suatu citra digital.

Arsitektur pelatihan jaringan ini terdiri dari masukan (x_i), layar tersembunyi (z_i), dan keluaran(y_i). Arsitektur pelatihan jaringan syaraf *backpropagation* yang akan dibangun pada sistem dapat dilihat pada Gambar 4.1 berikut ini.



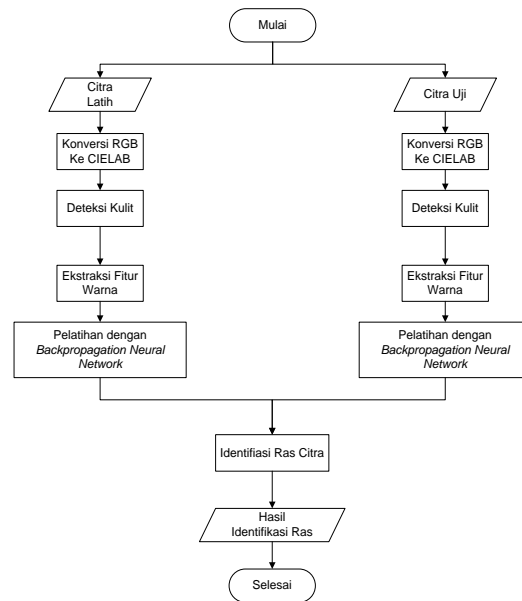
Gambar 4.1 Arsitektur Pelatihan Jaringan

Berdasarkan Gambar 4.1 di atas dapat diketahui bahwa pelatihan jaringan yang dibangun memiliki empat masukan, satu layar tersembunyi, dan satu keluaran.

B. Analisis Alur Kerja Sistem

Analisis alur kerja sistem merupakan analisis mengenai cara kerja dari suatu sistem yang dijelaskan secara terurut dimulai dari memasukan *input* oleh pengguna sampai sistem mengeluarkan hasil atau *output* melalui proses-proses yang

terdapat di dalam sistem. Analisis alur kerja sistem dari aplikasi ini ditunjukkan pada Gambar 4.2 di bawah ini.



Gambar 4.2 Diagram Alur Kerja Sistem

Berdasarkan diagram alur pada Gambar 4.2, masukan dari sistem ini terdiri dari dua masukan, yaitu citra latih dan citra uji. Citra latih adalah citra yang digunakan sebagai parameter karakteristik dalam mengidentifikasi ras manusia dari suatu citra. Sedangkan citra uji merupakan citra yang digunakan untuk menguji sistem dalam pengidentifikasi ras manusia berdasarkan warna kulit citra.

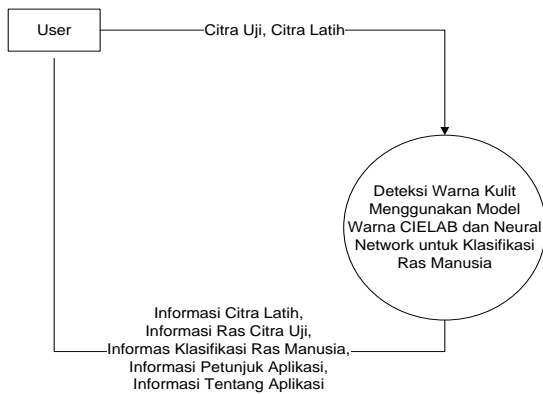
C. Data Flow Diagram (DFD)

Data Flow Diagram merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi yang dapat digunakan untuk penggambaran analisa sistem. DFD sistem ini dirancang dalam bentuk tiga level, yaitu diagram level 0, diagram level 1, dan diagram level 2 [7]. Berikut adalah penjabaran dari setiap level dari DFD.

1) Diagram Level 0

Diagram level 0 atau juga biasa disebut dengan diagram konteks merupakan diagram

tertinggi dari DFD. Diagram ini menggambarkan hubungan sistem dengan lingkungan di sekitar sistem. Diagram level 0 dari sistem ini ditunjukkan pada Gambar 4.3 berikut.



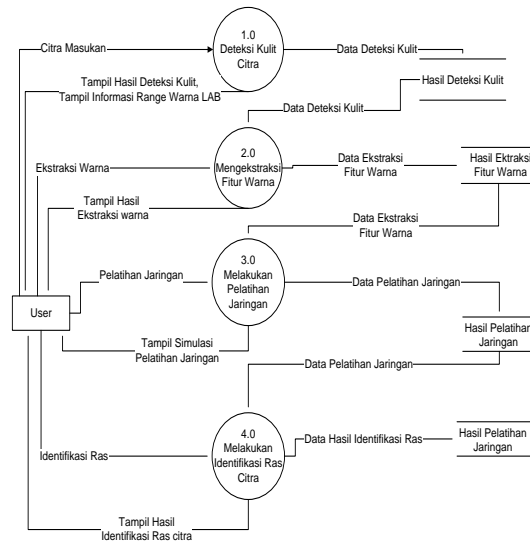
Gambar 4.3 Diagram Level 0 Aplikasi

Pada Gambar 4.3 diatas dapat dilihat bahwa terdapat satu proses dan satu entitas. Proses yang ada pada diagram level 0 ini adalah proses inti dari aplikasi yang akan dibangun. Pengguna dapat berinteraksi dengan sistem yaitu dengan memasukkan citra latih dan citra uji. Kemudian sistem juga berinteraksi dengan pengguna yaitu dengan memberikan informasi citra latih dari masukan citra latih yang telah dimasukan, informasi hasil identifikasi Ras dari citra uji, informasi tentang ras manusia, informasi tentang petunjuk aplikasi dan informasi tentang aplikasi.

2) Diagram Level 1

Diagram level 1 merupakan diagram turunan dari diagram konteks atau level 0. Diagram level 1 dari sistem ini ditunjukkan pada Gambar 4.4 berikut.

Berdasarkan Gambar 4.4, diagram level 1 pada sistem ini memiliki empat proses. Keempat proses tersebut akan dijelaskan sebagai berikut.



Gambar 4.4 Diagram Level 1 Aplikasi

a) Proses 1.0 Deteksi Kulit

Pada proses ini pengguna dapat berinteraksi dengan sistem dengan memasukkan citra kulit wajah manusia dari berbagai ras manusia yang akan dijadikan sebagai citra uji. Pada proses deteksi kulit ini, sistem akan melakukan proses pendeteksian bagian kulit dari citra. Dari hasil pendeteksian tersebut, kemudian sistem akan menampilkan informasi mengenai range warna dari setiap elemen warna $L^*a^*b^*$ dan menampilkan hasil pendeteksian bagian kulit dari citra.

b) Proses 2.0 Mengekstraksi Fitur Warna

Pada proses ini pengguna dapat berinteraksi dengan sistem dengan menekan tombol Ekstraksi Warna pada sistem. Kemudian sistem akan melakukan proses pengekstraksian fitur warna dari citra masukan. Dalam proses ekstraksi ini, dibutuhkan data hasil deteksi kulit yang didapatkan dari proses sebelumnya (proses 1.0) sehingga didapatkan hasil ekstraksi fitur warna yang akan digunakan dalam proses pengidentifikasina citra. Setelah itu sistem akan memberikan umpan balik kepada pengguna berupa informasi nilai hasil

ekstraksi warna yang merupakan nilai rata-rata dan nilai standar deviasi dari elemen a^* dan elemen b^* .

c) Proses 3.0 Melakukan Pelatihan Jaringan

Pada proses ini pengguna dapat berinteraksi dengan sistem dengan menekan tombol Latih Jaringan pada sistem. Kemudian sistem akan melakukan proses pelatihan jaringan berdasarkan jaringan yang telah dibangun pada sistem. Pada proses ini, sistem akan memberikan umpan balik kepada pengguna dengan menampilkan tampilan dari proses pelatihan jaringan.

d) Proses 4.0 Melakukan Identifikasi Ras Citra

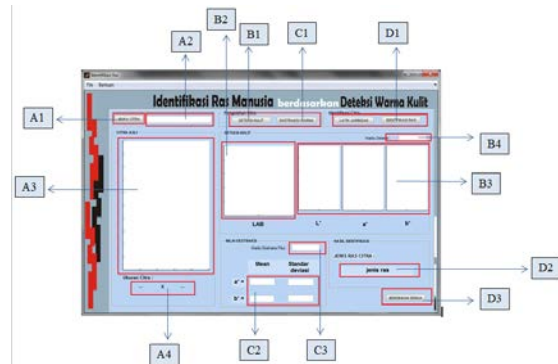
Pada proses ini pengguna dapat berinteraksi dengan sistem dengan menekan tombol Identifikasi Ras pada sistem. Kemudian sistem akan melakukan proses proses pengidentifikasian ras manusia dari citra masukan. Dalam proses identifikasi ini, dibutuhkan data hasil pelatihan jaringan yang telah dibuat untuk disimulasikan dengan nilai ekstraksi fitur warna dari citra uji yang dimasukkan sehingga ras dari citra uji dapat diidentifikasi oleh sistem. Setelah itu sistem akan memberikan umpan balik kepada pengguna berupa informasi hasil identifikasi Ras yang ada pada citra uji yang telah dimasukkan tersebut.

IV. PEMBAHASAN

Aplikasi deteksi warna kulit untuk identifikasi ras ini telah diimplementasikan menggunakan *software* MatLab R2013b versi 8.2.0.701 dengan *notebook* berspesifikasi Ram 2 GB dan *processor core i5*. Pengujian dilakukan dengan menggunakan citra dari dataset dan citra sendiri dengan format *.jpg.

A. Tampilan Antarmuka Identifikasi Ras

Halaman identifikasi ras adalah halaman yang muncul ketika memilih submenu 'Identifikasi Data Uji' pada halaman setiap halaman yang ditampilkan. Tampilan dari halaman identifikasi ras ini dapat dilihat pada Gambar 5.1 di bawah ini.



Gambar 5.1 Halaman Identifikasi Ras

Berdasarkan Gambar 5.1, pada halaman identifikasi ras ini terdiri dari lima *axes* yang digunakan untuk meletakkan citra, enam buah tombol yang dapat digunakan untuk melakukan proses deteksi dan proses identifikasi citra, serta delapan *field* yang digunakan untuk menampilkan informasi dari citra maupun informasi jenis ras. Setiap kode yang ada pada Gambar 5.1 memiliki fungsi dan arti yang berbeda antara satu dengan yang lain [8]. Berikut uraian dari masing-masing kode yang terdapat pada Gambar 5.2.

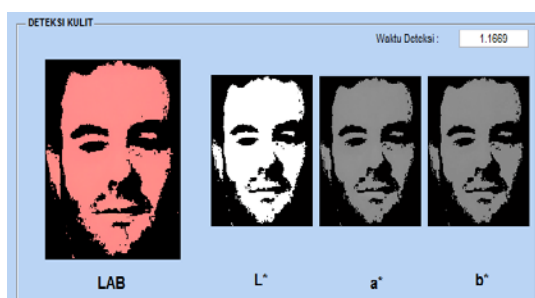
Kode A merupakan kode yang diberikan untuk proses yang dilakukan pertama kali dalam menjalankan aplikasi. Kode A1 merupakan tombol 'buka citra' yang digunakan untuk memasukan citra uji yang akan diidentifikasi, kode A2 merupakan *field* yang digunakan untuk menampilkan informasi nama citra masukan, kode A3 merupakan *axes* yang digunakan untuk menampilkan citra masukan, sedangkan kode A4 merupakan *field* yang digunakan untuk menampilkan informasi mengenai ukuran citra

masukan. Tampilan dari proses pertama pada pengoperasian aplikasi ditunjukkan pada Gambar 5.2 di bawah ini:



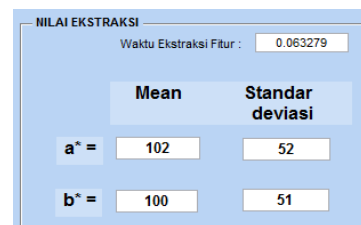
Gambar 5.2 Tampilan Citra Masukan

Kode B merupakan kode yang diberikan untuk proses kedua yang dilakukan pada aplikasi. Kode B1 merupakan tombol 'Deteksi Kulit' yang digunakan untuk melakukan proses deteksi kulit dari citra masukan, kode B2 merupakan *axes* yang digunakan untuk menampilkan citra hasil deteksi kulit dengan model warna LAB, kode B3 terdiri dari tiga *axes* untuk menampilkan hasil deteksi kulit dengan yang dipisahkan menjadi tiga bagian yaitu komponen L^* , a^* , dan b^* , sedangkan kode B4 merupakan *field* yang digunakan untuk menampilkan informasi waktu pendeteksian kulit pada citra masukan. Tampilan dari kode B ditunjukkan pada Gambar 5.3 di bawah ini:



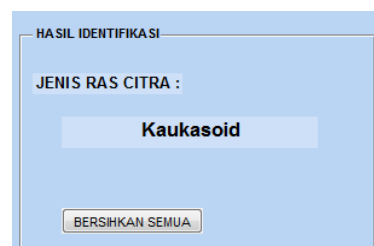
Gambar 5.3 Tampilan Hasil Deteksi Kulit

Kode C merupakan kode yang diberikan untuk proses ketiga yang dilakukan pada aplikasi. Kode C1 merupakan tombol 'Ekstraksi Warna' yang digunakan untuk melakukan proses ekstraksi warna dari elemen a^* dan b^* dari citra hasil deteksi kulit, kode C2 terdiri dari empat *field* yang menampilkan informasi nilai *mean* dan nilai *standar deviasi* dari elemen a^* dan b^* , sedangkan kode C3 merupakan *field* yang digunakan untuk menampilkan informasi waktu pendeteksian kulit pada citra masukan. Tampilan dari kode C ditunjukkan pada Gambar 5.4 di bawah ini:



Gambar 5.4 Hasil Ekstraksi Citra

Kode D merupakan kode yang diberikan untuk proses kelima yang dilakukan pada aplikasi setelah melakukan proses latih jaringan pembelajaran. Kode D terdiri dari tiga bagian yaitu D1, D2, dan D3. Kode D1 merupakan tombol 'Identifikasi Ras' yang digunakan untuk melakukan proses identifikasi ras dari citra masukan, kode D2 merupakan *field* yang digunakan untuk menampilkan hasil identifikasi ras dari citra masukan, sedangkan kode D3 merupakan tombol 'Bersihkan Semua' yang digunakan untuk membersihkan tampilan antarmuka identifikasi ras. Tampilan dari kode D ditunjukkan pada Gambar 5.5 di bawah ini:



Gambar 5.5 Tampilan Hasil Identifikasi Ras Citra

B. Pengujian Sistem

Hasil pengujian identifikasi ras manusia yang dilakukan sistem merupakan hasil pembulatan angka dari hasil simulasi yang dilakukan pada sistem. Jika *output* hasil simulasi menghasilkan nilai $> 0,5$ dan $< 1,5$ maka akan dibulatkan menjadi 1, sehingga akan diidentifikasi sebagai ras Kaukasoid. Jika *output* hasil simulasi menghasilkan nilai $> 1,5$ dan $< 2,5$ maka akan dibulatkan menjadi 2, sehingga akan diidentifikasi sebagai ras Mongoloid. Jika *output* hasil simulasi menghasilkan nilai $> 2,5$ dan $< 3,5$ maka akan dibulatkan menjadi 3, sehingga akan diidentifikasi sebagai ras Negroid. Pengujian yang dilakukan pada sistem adalah sebagai berikut:

1. Hasil pengujian optimasi jaringan pelatihan

Pengujian ini dilakukan untuk mendapatkan arsitektur jaringan pelatihan *backpropagation* yang memiliki nilai akurasi terbaik dibandingkan dengan arsitektur lainnya. Optimasi jaringan yang dilakukan adalah dengan menggunakan kombinasi jumlah *neuron* dan kombinasi fungsi aktivasi pada layar tersembunyi. Kombinasi *neuron* yang digunakan adalah 20, 30, dan 40, sedangkan kombinasi fungsi aktivasi yang digunakan adalah *logsig* dan *tansig*. Tabel 5.1 berikut ini merupakan tabel hasil pengujian kombinasi optimasi yang dilakukan.

Tabel 5.1 Pengujian kombinasi optimasi pelatihan jaringan

Aktivasi	Neuron	Berhasil Dikenali	Akurasi (%)
Logsig	20	26	86,67
	30	28	93,33
	40	24	80
Tansig	20	26	86,67
	30	27	90
	40	26	86,67






Dari tabel 5.1 di atas diketahui dari kombinasi optimasi yang dilakukan pada pelatihan jaringan,




kombinasi yang memiliki nilai akurasi terbaik adalah pelatihan dengan jumlah neuron sebanyak 30 pada layar tersembunyi dan dengan menggunakan fungsi aktivasi *logsig* yang berhasil mengidentifikasi ras pada 28 data citra dengan benar dan memiliki nilai akurasi sebesar 93,33%.

2. Hasil pengujian identifikasi menggunakan citra uji dataset SFA

Pengujian ini dilakukan dengan menguji aplikasi dalam mengidentifikasi ras pada citra masukan yang berasal dari Dataset SFA. Citra dari dataset yang digunakan sebagai citra uji yang dipilih secara acak tanpa memperhatikan pencahayaan dan sudut pengambilan dari gambar. Jumlah citra yang digunakan dalam pengujian adalah sebanyak 30 citra yang terdiri dari 10 ras kaukasoid, 10 ras mongoloid, dan 10 ras negroid. Hasil dari pengujian yang dilakukan ini dapat dilihat pada tabel 5.2 berikut ini.

Tabel 5.2 Pengujian menggunakan citra uji dari dataset SFA

Citra Masukan	Manual	Target	Hasil Simulasi	Kelompok Ras	Ket.
	Kaukasoid	1	1,3039	Kaukasoid	Benar
	Kaukasoid	1	1,2057	Kaukasoid	Benar
	Kaukasoid	1	1,4422	Kaukasoid	Benar
	Mongoloid	2	2,0111	Mongoloid	Benar
	Mongoloid	2	2,0710	Mongoloid	Benar

	Negroid	3	3,1611	Negroid	Benar
	Negroid	3	2,5208	Negroid	Benar
	Negroid	3	3,2240	Negroid	Benar

Berdasarkan tabel 5.2 dapat diketahui bahwa hasil identifikasi ras dari 30 citra uji yang dimasukan terdapat 28 citra yang diidentifikasi dengan benar sesuai dengan target. Maka tingkat akurasi yang diperoleh dari pengujian ini adalah sebesar 93,33%. Tingkat akurasi ini diperoleh dengan menggunakan persamaan (5.1) seperti berikut.

$$Akurasi (\%) = \frac{28}{30} \times 100 \% = 93,33\%$$









Kesalahan dalam pengidentifikasian citra terjadi karena pada citra tersebut memiliki nilai warna kulit yang berbeda dengan citra lain dari ras yang sama. Hal ini dapat diakibatkan oleh faktor pencahayaan maupun cara pengambilan gambar yang dilakukan, sehingga citra memiliki karakteristik warna CIELAB diluar range warna kulit pada ras yang dimilikinya.

3. Hasil pengujian identifikasi menggunakan citra uji dokumen sendiri

Pengujian ini dilakukan dengan menguji aplikasi dalam mengidentifikasi ras pada citra masukan yang diambil dokumen pribadi yang berasal dari pencarian di internet dan pengambilan sendiri menggunakan kamera *Samsung SM-N750* 8 Megapiksel. Jumlah citra yang digunakan adalah sebanyak 15 citra. Hasil dari pengujian yang

dilakukan ini dapat dilihat pada tabel 5.3 berikut ini.

Tabel 5.3 Pengujian dengan menggunakan citra uji foto sendiri

Citra Masukan	Manual	Target	Hasil Simulasi	Kelompok Ras	Ket.
	Kaukasoid	1	0,9627	Kaukasoid	Benar
	Kaukasoid	1	1,3129	Kaukasoid	Benar
	Mongoloid	2	1,8690	Mongoloid	Benar
	Mongoloid	2	1,6571	Mongoloid	Benar
	Mongoloid	2	1,5586	Mongoloid	Benar
	Mongoloid	2	1,9007	Mongoloid	Benar
	Negroid	3	2,8387	Negroid	Benar
	Negroid	3	2,9582	Negroid	Benar

Berdasarkan tabel 5.3 di atas dapat dilihat bahwa hasil identifikasi ras dari 15 citra uji yang dimasukan terdapat 14 citra yang diidentifikasi dengan benar sesuai dengan target. Maka tingkat akurasi yang diperoleh dari pengujian ini adalah sebesar 93,33%. Tingkat akurasi ini diperoleh

dengan menggunakan persamaan (5.1) seperti di bawah ini:

$$\text{Akurasi (\%)} = \frac{14}{15} \times 100 \% = 93,33\%$$

Kesalahan dalam pengidentifikasian citra terjadi karena pada citra tersebut memiliki nilai warna kulit yang berbeda dengan citra lain dari ras yang sama. Hal ini dapat diakibatkan oleh faktor eksternal yang mengakibatkan adanya perubahan warna kulit, seperti dikarena oleh faktor pencahayaan pada saat pengambilan citra dan perbedaan jenis kamera yang digunakan sehingga range warna CIELAB yang dikenali pada citra mengalami perbedaan dari citra lain dari ras yang sama.

V. KESIMPULAN

Berdasarkan hasil dan pembahasan yang telah diuraikan sebelumnya, maka kesimpulan yang dapat diambil adalah sebagai berikut:

1. Penelitian ini telah menghasilkan aplikasi deteksi warna kulit untuk identifikasi ras manusia pada citra digital dengan studi kasus ras yaitu, Kaukasoid, Mongoloid, dan Negroid berbasis desktop dengan menggunakan model warna *CIELAB* dan *Backpropagation Neural Network*.
2. Model warna *CIELAB* dan *Backpropagation Neural Network* yang diimplementasikan pada aplikasi ini menghasilkan tingkat akurasi sebagai berikut:
 - a. Pengujian optimasi jaringan pelatihan dengan menggunakan jumlah *neuron* 30 dan fungsi aktivasi *logsig* memiliki keakurasian paling baik dari pengujian optimasi lainnya, yaitu memiliki nilai akurasi sebesar 93,3%
 - b. Pengujian dengan menggunakan 30 citra masukan yang berasal dari dataset,

memiliki tingkat keakurasian sebesar 93,33 %.

- c. Pengujian dengan menggunakan 15 citra masukan yang berasal dari foto sendiri serta pengambilan dari internet, memiliki tingkat keakurasian sebesar 93,33 %.

VI. SARAN

Berdasarkan hasil dan pembahasan yang telah diuraikan sebelumnya, maka saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Kekurangan dari hasil deteksi warna kulit dengan menggunakan model warna *CIELAB* dan *Backpropagation Neural Network* yang telah diimplementasikan dalam aplikasi adalah adanya kesalahan dalam pendeteksian kulit apabila citra masukan memiliki latar belakang gambar dengan warna yang memiliki kesamaan dengan warna kulit manusia, sehingga latar belakang dari citra masukan juga terdeteksi sebagai warna kulit dan mempengaruhi sistem pada proses pengidentifikasian jenis ras manusia dari citra. Sehingga kedepannya pengembangan aplikasi dapat difokuskan untuk meminimalisir kesalahan pendeteksian tersebut, misalnya menggunakan metode yaitu *Gaussian Mixture Feature* pada proses pendeteksian bagian kulit manusia.
2. Pada saat pengambilan citra sebaiknya pengaruh pencahayaan harus diperhatikan, Karena pencahayaan yang terlalu gelap atau terang akan mempengaruhi dalam proses pengenalan warna kulit, sehingga proses pengidentifikasian citra tidak dapat dilakukan secara maksimal. Sehingga untuk memperbaiki pengaruh pencahayaan dalam pengenalan warna kulit pada citra dapat dilakukan dengan menormalisasi pencahayaan pada citra.

REFERENSI

- [1] Murinto, E. Aribowo and W. Nurhidayati, "Deteksi Jenis Warna Kulit Wajah untuk Klasifikasi Ras Manusia Menggunakan Transformasi Warna," Universitas Ahmad Dahlan, Yogyakarta, 2008.
- [2] C. R. Ember and M. Ember, Anthropology 4th Edition, New York: Hunter College of the City University of New York, 2000.
- [3] A. Mall and S. G. Mazumdar, "Skin Tone Based Face Recognition and Training using Neural Network," *Internasional Journal of Emerging Technology and Advanced Engineering*, pp. 1-5, 2012.
- [4] A. Kadir and A. Susanto, Teori dan Aplikasi Pengolahan Citra, Yogyakarta: Penerbit Andi, 2013.
- [5] J. J. Siang, Jaringan Syaraf Tiruan dan Pemrogramannya dengan Matlab, Yogyakarta: Penerbit Andi, 2005.
- [6] N. Nurmila, E. A. Sarwoko and A. Sugiharto, "Algoritma Backpropagation Neural Network untuk Pengenalan Pola Karakter Huruf Jawa," *Jurnal Masyarakat Informatika Volume 1*, 2010.
- [7] A. Dennis, B. H. Wixom and Tegarden, System Analysis and Design with UML Version 2.0 An Object-Oriented Approach Second Edition, United States of America: John Wiley & Sons, 2005.
- [8] E. Paulus and Y. Nataliani, Gui Matlab, Yogyakarta: Penerbit Andi, 2007.