**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang**

Pada saat ini perkembangan di bidang teknologi dan informasi (TI) sangat pesat. Perkembangan ini di manfaatkan di berbagai bidang untuk membantu manusia dalam mengerjakan pekerjaannya dan juga dapat menjadi hiburan di berbagai kalangan, mulai dari anak-anak hingga dewasa. Salah satu bidang TI yang berkembang saat ini adalah pengolahan citra. Terdapat berbagai jenis aplikasi untuk pengolahan citra seperti *Photoshop, Corel draw, dan Paint*. Aplikasi-aplikasi pengolahan citra dapat digunakan untuk memperbaiki kualitas gambar, mengubah ukuran gambar serta dapat memodifikasi gambar.

Citra merupakan fungsi kontinyu dengan intensitas cahaya pada bidang dua dimensi. Citra harus direpresentasikan secara numerik dengan nilai diskrit agar dapat diolah dengan komputer. Salah satu domain pengolahan citra adalah mengklasifikasi citra dikelompok tertentu, misalmya bertujuan untuk menentukan warna kulit manusia.

Mengklasifikasi warna kulit ini dari geografis berbeda, seperti kulit orang Asia, kulit orang Afrika, kulit orang Eropa dan sebagainya. Klasifikasi warna kulit digunakan untuk mengetahui ras manusia. Hal ini dapat digunakan mengidentifikasi manusia dalam bidang kriminal untuk mengetahui asal usul korban tindak kejahatan, dan dapat digunakan untuk penentuan keputusan suatu penyakit genetik pada bidang kesehatan.

Untuk mengklasifikasi warna kulit, diperlukan suatu metode yang dapat mendefinisikan suatu piksel dalam membedakan warna kulit. Salah satu metode yang dapat digunakan adalah metode *K-Nearest Neighbour* (KNN). KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap *training data* yang memiliki banyak *noise* dan efektif apabila *training data*-nya besar (sikki, 2009). Oleh karena itu, pada penelitian ini akan diteliti tentang klasifikasi warna kulit dengan menggunakan algoritma *K-Nearest Neighbour*.

**1.2 Perumusan Masalah**

Bagaimana efektivitas algoritma *K-Nearest Neighbor* (KNN) untuk mengklasifikasi warna kulit manusia.

**1.3 Batasan Masalah**

Tugas akhir ini mempunyai batasan-batasan dalam menangani pengolahan citra. Batasan masalah dalam program ini secara garis besarnya adalah:

* 1. Pada tugas akhir ini, algoritma *K-Nearest* Neighbour (KNN) untuk mendeteksi dan mengklasifikasi warna kulit manusia di implementasikan dengan menggunakan bahasa JAVA.
  2. Data citra yang berupa foto sampel dari masing-masing warna kulit dan non kulit diambil dari internet.
  3. Klasifikasi warna kulit hanya dikelompokkan kedalam 5 warna seperti hitam pekat, hitam, coklat, kuning langsat dan putih dengan sampel sebanyak 100 gambar dari masing-masing warna dan 100 sampel dari citra bukan kulit.
  4. Format gambar yang digunakan adalah gambar berformat jpeg.

**1.4 Tujuan Penelitian**

Tujuan yang dapat dirumuskan dari penelitian Tugas Akhir ini adalah :

* 1. Untuk mengetahui tingkat keberhasilan algoritma KNN dalam pengklasifikasian 5 warna kulit yaitu hitam pekat, hitam, coklat, kuning langsat, dan putih.
  2. Aplikasi yang dibuat dapat digunakan untuk mengklasisfikasikan warna kulit hitam pekat, hitam, coklat, kuning langsat, dan putih.

**1.5 Manfaat Penelitian**

Adapun manfaat dari penelitian Tugas Akhir ini adalah:

* 1. Untuk menambah ilmu pengetahuan mengenai pengolahan citra dan algoritma K- Nearest Neighbour.
  2. Untuk pengklasifikasian beberapa warna kulit.
  3. Menjadi bahan referensi atau acuan di Universitas Mataram pada umumnya, di Fakultas Teknik khususnya untuk penelitian yang berhubungan dengan pengolahan citra.

**1.6 Sistematika Penulisan**

Adapun sistematika penulisan laporan Tugas Akhir ini adalah sebagai berikut :

1. Bab I Pendahuluan : Merupakan bagian yang berisikan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.
2. Bab II Teori Penunjang : Pada bab ini berisi tentang Tinjauan Pustaka dan Dasar Teori yang mendukung penyusunan Tugas Akhir ini*.*
3. Bab III Metodologi Penelitian : Merupakan bagian-bagian yang berisikan tentang alat dan bahan, dan langkah – langkah penelitian serta rancangan sistem yang akan dikembangkan sebagai Tugas Akhir.
4. Bab IV Hasil Dan Pembahasan : Pada BAB ini memuat tentang analisa dan pembahasan hasil penelitian yang telah dilakukan.
5. Bab V Kesimpulan dan Saran : Memuat tentang kesimpulan dan saran-saran berdasarkan hasil pembahasan yang diperoleh.

**BAB II**

**TINJAUAN PUSTAKA DAN DASAR TEORI**

**2.1 Tinjauan Pustaka**

Dalam penelitian tentang *human skin color detection in RGB space with Bayesian estimation with beta mixture model* *(BMM)*, deteksi warna kulit/ bukan kulit dari manusia berdasarkan nilai piksel pada RGB (*Red*, *Green*, *Blue*), menggunakan metode Bayesian untuk mengestimasi pengklasifikasian kulit dan bukan kulit dengan parameter BMM (Ma dan Arne, 2010). Metode pada penelitian ini cukup baik digunakan untuk hasil pendeteksian.

Dari penelitian tentang pengklasifikasian warna kulit berbasis piksel menggunakan jaringan syaraf tiruan (JST), untuk mengklasifikasi kulit/non-kulit dari manusia dengan menggunakan metode *back propagation* (BP) (Susanti, 2014). Metode pada penelitian ini efektif untuk gambar warna non-kulit yang tidak mirip dengan warna kulit.

Penelitian tentang *A survey of skin-color modeling and detection methods*, didapatkan kesimpulan bahwa *Gaussian modeling* efektif untuk penentuan space warna, *model histogram* dengan *native bayes classifiers* efektif untuk pendeteksian kulit karena memiliki database yang besar (Kakumanu dkk., 2007). Teknik ini baik untuk pendeteksian kulit.

Penelitian pengenalan wajah menggunakan *K- Nearest Neighbour* dengan proproses transpormasi wavelet, untuk pendeteksian citra wajah (Sikki, 2009). Metode pada penelitian ini cukup baik digunakan untuk hasil pendeteksian.

Penelitian tentang deteksi dan klasifikasi citra berdasarkan warna kulit menggunakan *Hue Saturation Value* (HSV), mengemukakan model HSV digunakan untuk segmentasi warna kulit manusia dengan membandingkan piksel sampel image yang akan dideteksi. HSV juga digunakan untuk mengklasifikasi citra khususnya dalam klasifikasi gambar pornografi. Dengan memberikan rentang piksel 0%-20%, 21%-50%, dan 51%-100%. Hasil deteksi dikelompokkan dalam tiga kategori yaitu gambar bukan pornografi, gambar semi pornografi dan gambar pornografi (Wibowo, 2011). Namun latar gambar yang memiliki warna yang sama seperti warna kulit akan menyebabkan piksel kulit bertambah.

**2.2 Dasar Teori**

2.2.1 Operasi Pengolahan Citra

Secara harfiah citra (*image*) merupakan gambar pada bidang du dimensi (dwimitra). Secara matematis, citra merupakan fungsi menerus (*continue*) pada bidang dwimitra dari intensitas cahaya. Sumber cahaya menerangi objek, sebagian dari berkas cahaya tersebut, dipantulkan kembali oleh objek. Pantulan cahaya ini ditangkap oleh alat-alat optic, seperti mata pada manusia, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam (Munir, 2004).

Pangolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik.

Umumnya operasi-operasi pada pengolahan citra diterapkan pada citra bila (Munir, 2004) :

* 1. Perbaikan atau modifikasi citra perlu dilakukan untuk menonjolkkan beberapa aspek informasi yang terkandung didalam citra atau meningkatkan kualitas penampakan,
  2. Elemen didalam citra pelu diukur, dikelompokkan, atau dicocokkan,
  3. Sebagian citra perlu digabung dengan citra yang lain.

Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal komputer). Teknik-teknik pengolahann citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, citra keluaran (*output*) memiliki kualitas lebih baik daripada citra masukan (*input*).

suatu citra harus direpresentasikan secara numeric supaya dapat diolah dengan computer digital dengan nilai-nilai diskrit. Represenasi citra dari fungsi malar (kontinu) menjadi nilai diskrit disebut digitalisasi. Citra yang dihasilkkan disebut dengan citra digital (*digital image*). Pada umumnya citra digital berbentuk empat persegi panjang, dan dimensi ukurannya dinyatakan sebagai tinggi *x* lebar (atau lebar *x* panjang).

Citra digital yang tingginya *N*, lebarnya *M*, dan memiliki *L* derajat keabuan dapat dianggap sebagai fungsi (Munir, 2004):

(2-1)

2.2.2 Warna

Warna dari sebuah objek yang diterima ditentukan oleh warna sinar yang dipantulkan oleh objek tersebut. Sinar tampak (*visible spectrum*) adalah warna yang direspon oleh mata dengan panjang gelombang berkisar dari 400 (biru) sampai 700nm (merah).

2.2.2.1 Model warna RGB

Kombinasi cahaya dengan panjang gelombang berbeda merupakan warna-warna yang diterima oleh mata. Menurut penelitian, bahwa kombinasi warna yang memberikan rentang warna paling lebar adalah *red* (R), *green* (G), dan *blue* (B). Warna-warna ini disebut warna pokok (*primaries*), dan sering disingkat sebagai warna dasar RGB. Warna-warna lain dapat diperoleh dengan mencampurkan ketiga warna pokok tersebut. Sesuai dengan teori Young (1802) yang menyatakan bahwa sembarang warna dapat dihasilkan dari percampuran warna-warna pokok dengan persentase tertentu (Munir, 2004):

(2-2)

2.2.2.2 Model Warna YCbCr

Ruang warna YCbCr disebut juga ruang warna CCIR 601 (*International Radio Consultative Commite*). Model warna ini banyak digunakan pada video digital karena dikembangkan untuk mengantisipasi perkembangan informasi berbasis video. Secara umum model warna ini merupakan bagian dari ruang warna transmisi video dan televisi. Model warna YCbCr memisahkan nilai RGB menjadi informasi *luminance* dan *chominance* yang berguna untuk aplikasi kompresi. Transformasi RGB ke YCbCr dilakukan dengan formulasi operasi matriks pada persamaan (2-3) dan (2-4) sebagai berikut :

Y = 0.299900R + 0.58700G + 0.11400B

Cb=-0.16874R-0.33126G-0.50000B (2-3)

Cr = 0.50000R - 0.41869G - 0.08131B

R = 1.00000Y + 1.40200Cr

G = 1.00000T - 0.34414Cb (2-4)

B = 1.00000Y + 1.77200Cb

Formula konversi RGB – YcbCr dalam format lain ditunjukkaan sebagai berikut:

(2-5)

Transformasi warna dari basis CIE RGB ke CIE XYZ dapat dilakukan sebagai berikut, diberikan triplet *RGB* untuk piksel *I*, maka triplet *XYZ* dihitung dengan :

(2-6)

Transformasi sebaliknya dari *CIE XYZ* ke *CIE RGB* dapat dilakukan dengan persamaan :

(2-7)

Model warna RGB dan XYZ yang dikemukakan diatas adalah bakuan dari CIE. Ada juga model warna yang diusulkan untuk *flatform* perangkat keras tertentu untuk menampilkan gambar. Misalnya **Nation Television System Committee (NTSC)** menggunakan model warna RGB khusus untuk menampilkan citra berwarna pada layar CRT. Format televisi di Amerika Serikat menggunakan format NTSC. Format ini dapat memisahkan data *grayscale* dari data warnanya, sehingga sinyal yang sama dapat digunakan baik untuk layar layar hitam putih maupun berwarna (Susanti, 2014).

2.2.3 Warna Kulit Manusia

Kulit adalah organ yang paling terlihat yang digunakan untuk membedakan seseorang dengan yang lainnya. Untuk mengidentifikasi ras dan keturunan orang dapat dilakukan membedakan jenis warna kulit. bangsa Indonesia atau Melayu pada umumnya memiliki kulit berwarna cokelat sawo matang atau kuning langsat, bangsa Eropa berkulit putih, bangsa China berkulit kuning, dan Afrika berkulit hitam.

Jenis warna kulit manusia dikoelompokkan menjadi lima jenis oleh Thomas B. Fitzpatrick pada penelitiannya. Setiap warna kulit memiliki karakter serta kelebihan dan kekurangan masing-masing. Kelima jenis tersebut adalah very fair, fair, medium, olive, serta brown atau black.

*Very fair* merupakan jenis warna kulit yang memiliki tingkat intensitas warna putih yang sangat terang dan cenderung pucat. Warna kulit ini meski terkena matahari tidak akan mencoklat atau menghitam. Masyarakat keturunan ras Eropa dan Australia memiliki kulit dengan warna jenis ini. Jenis warna kulit Fair cenderung lebih pink dan banyak muncul di jenis kulit bangsa Jepang dan Cina. Warna kulit fair terkadang bisa saja berubah menjadi gelap. Orang keturunan Indian Meksiko dan Indonesia pada umumnya memiliki warna kulit medium. Akan cukup mudah berubah menjadi gelap bila terpapar sinar matahari. Warna kulit olive walaupun jarang terpapar sinar matahari, warnanya akan tetap cokelat. Pada bangsa Afrika dan India warna kulit brown atau black banyak didapatikan. Warna kulit ini akan selalu menggelap atau hitam meskipun meskipun tidak pernah terkena sinar matahari langsung (Novintaclenesse, 2015).

2.2.4 Confusion Matrix

*Confusion matrix* berisi informasi tentang klasifikasi aktual dan prediksi yang dilakukan oleh sistem klasifikasi (Hamilton, 2012). Kinerja sistem menggunakan data dalam matriks untuk dievaluasi. Tabel berikut menunjukkan *confusion matrix.*

Tabel 2.1. *confusion matrix*

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Prediksi | |
| Negative | Positive |
| Aktual | Negative | a | b |
| Positive | c | d |

Berikut merupakan penjelasan dari tabel *confusion matrix* diatas :

a adalah jumlah prediksi yang benar pada kelas negatif,

b adalah jumlah prediksi yang salah pada kelas positif,

c adalah jumlah yang salah prediksi pada kelas negatif, dan

d adalah jumlah prediksi yang benar pada kelas positif.

Berikut merupakan Rumus untuk menhitung akurasi, *false positive rate* (FPR)*,* dan *false negative rate*.

(2-9)

(2-10)

(2-11)

*Accuracy* : Persentase keberhasilan yang didapatkan.

FPR(*False Positive Rate*): proporsi dari nilai positif yang salah, di klasifikasikan sebagai nilai negatif.

FNR(*False Negative Rate*): proporsi dari nilai negatif yang salah diklasifikasikan sebagai nilai positif.

2.2.5 Algoritma KNN

KNN adalah suatu metode dimana hasil dari *query instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori pada KNN dengan menggunakan algoritma *supervised* (Sikki, 2009). Adapun tujuan dari algoritma ini yaitu mengklasifikasikan obyek baru berdasarkan atribut dan *training sample*. Diberikan titik *query*, akan ditemukan sejumlah *k* obyek atau (titik *training*) yang paling dekat dengan titik *query*. Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari *k* obyek. Algoritma KNN menggunakan ketetanggaan sebagai nilai prediksi dari *query instance* yang baru.

Untuk menentukan KNN-nya, algoritma KNN bekerja berdasarkan jarak terpendek dari *query instance* ke training sample. Training sample diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi training sample. Sebuah titik pada ruang ini ditandai kelas *c*. jika kelas *c* merupakan klasifikasi yang paling banyak ditemui pada *k* buah tetangga terdekat dari titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan Mahalanobis Distance yang direpresentasikan sebagai berikut (Ker, 2010) :

(2-12)

Dimana :

D : *Distance*

Xi : Data uji ke-i

µ: Mean data latih

T : *Transpose*

C-1 : Invers Matriks kovarian

Matriks kovarian tersebut direpresentasikan sebagai berikut (Ker, 2010) :

(2-13)

C : Kovarian

: Data latih window ke-i

: Rata-rata data

n : Jumlah data

T : Transpose

Algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data *training* sample pada fase *training*. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk testing data (yang klasifikasinya tidak diketahui). Seluruh vektor training sample dari jarak vektor baru dihitung dan sejumlah *k* buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik tersebut.

Nilai *k* yang terbaik pada algoritma ini tergantung dari data. Secara umum, nilai *k* yang akan mengurangi efek noise pada klasifikasi terdapat pada *k* tertinggi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur.

Ada atau tidaknya fitur-fitur yang tidak relevan sangat mempengaruhi ketepatan algoritma KNN, atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik (Sikki, 2009).

Penggunaan bobot untuk menghitung kandidat kelas yang sebaiknya diambil oleh data uji dari K-tetangga terdekat. Bobot dari tetangga terdekat dihitung dengan formula (Prasetyo, 2012):

(2-14)

*d(X’,X)* (2-15)

(2-16)

Formula yang biasa digunakan untuk memasukkan sejumlah data ke kelas *Y’* adalah :

(2-17)

Keterangan :

*X’*: data uji

*Y’* : label kelas data uji yang belum diketahui

*d(X’,X)* : jarak diantara data uji *Z* ke setiap vector data latih, kemudian di simpan di dalam *D*.

KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap banyak *noise yang* dimiliki *traininig data* dan efektif apabila *training data*-nya besar. Sedangkan, kelemahan KNN yaitu perlunya penentuan nilai dari parameter *k* (jumlah dari tetangga terdekat) pada KNN, biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap *query instance* pada keseluruhan *training sample*, dan *training* berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik (Sikki, 2009).

Nearest neighbor merupakan teknik klasifikasi yang sederhana, tetapi mempunyai hasil kerja yang cukup bagus. Meskipun begitu, K-NN juga mempunyai kelebihan dan kekurangan. Beberapa karakteristik K-NN adalah sebagai berikut :

1. K-NN merupakan algoritma yang menggunakan seluruh data latih untuk melakukan proses klasifikasi (*complete storage*). Sehingga proses prediksi yang sangat lama untuk data dalam jumlah yang sangat besar.
2. Algoritma K-NN tidak membedakan setiap fitur dengan suatu bobot seperti pada *Artificial Neural Network* (ANN) yang berusaha menekan fitur yang tidak mempunyai kontribusi terhadap klasifikasi menjadi 0 pada bagian bobot. K-NN tidak memiliki bobot untuk masing-masing fitur.
3. Hal yang rumit adalah menentukan nilai *K* yang paling sesuai.
4. Karena K-NN pada prinsipnya memilih tetangga terdekat, parameter jarak juga penting untuk dipertimbangkan sesuai dengan kasus datanya. *Euclidean* sangat cocok untuk menggunakan jarak terdekat (lurus) antara dua data, tetapi Manhattan sangat teguh (*Robust*) untuk mendeteksi *outlier* dalam data (Prasetyo, 2012).

Berikut merupakan *flow chart* dari algoritma KNN



Gambar 2.1. *Flow Chart* Algoritma KNN

**BAB III**

**METODOLOGI PENELITIAN**

**3.1 Alat dan Bahan**

Alat dan Bahan yang digunakan adalah sebagai berikut:

1. *Notebook* Samsung NP905S3G-K02ID *Quad-Core Processor* DDR3 4 GB.
2. *Software java Netbeans* IDE 8.0. untuk pembuatan program.
3. *Software paint* untuk pengeditan beberapa citra yang akan digunakan seperti *cropping*, *resize* dan lainnnya.
4. Microsoft Windows 8 Sebagai Sistem Operasi.
5. Citra kulit dan bukan kulit.

**3.2 Proses Penelitian**

Pada bab ini akan dibahas mengenai bagaimana alur dari proses penentuan warna kulit dari suatu citra adalah sebagai berikut:

**3.2.1 Studi literatur**

Metode ini digunakan untuk mendapatkan berbagai informasi secara garis besar yang digunakan sebagai acuan dalam menyelesaikan permasalahan. Metode ini dilakukan dengan mempelajari buku-buku, literatur, jurnal, dan mengakses beberapa situs yang berhubungan dengan permasalahan yang diangkat agar ditemukan gambaran solusi dari masalah yang diteliti dari tugas akhir ini.

**3.2.2 Alur Penelitian**

Pada Gambar 3.1 pada alur penelitian dapat diketahui bahwa dari penelitian ini, dimulai dengan pengumpulan data dan literatur, dilanjutkan dengan perancangan sistem yang akan dibangun, pembuatan program serta pengujian program. Apabila program belum berhasil, maka proses akan kembali ke perancangan sistem yang di buat. Jika program yang di buat berhasil, maka akan dilanjutkan ke pembuatan laporan tugas akhir ini.

Pada blok pengumpulan data dan literatur, dilakukan pengumpulan data dan informasi yang digunakan sebagai acuan dalam menyelesaikan permasalahan. Blok selanjutnya yaitu perancangan sistem, pada blok ini dilakukan perancangan terhadap sistem yang akan di bangun. Setelah sistem telah dirancang, maka tahap selanjutnya yaitu merealisasikan rancangan yang telah dibuat, dengan membuat program. Apabila program sudah dibuat, maka program tersebut di uji, apakah sudah sesuai dari yang diharapkan atau belum. Jika program tersebut belum sesuai, maka proses akan kembali lagi ke proses perancangan sistem, tetapi jika program tersebut telah sesuai yang diharapkan maka akan dilanjutkan dengan pembuatan laporan mengenai penelitian yang telah dikerjakan.



Gambar 3.1 Diagram Alur penelitian.

**3.2.3 Sistem Klasisfikasi Kulit**

Bagan sistem yang akan dibangun adalah sebagai berikut:

pelatihan pengujian



Gambar 3.2 Bangan Sistem Klasifikasi Kulit

Dari Gambar 3.2 terdapat 2 proses utama yaitu pelatihan dan pengujian.

Proses pelatihan meliputi:

* Menyiapkan citra 5 warna kulit dan non kulit untuk dilatih.
* Masukan citra yang berupa nilai RGB dikonversi ke nilai YCbCr.
* Melakukan sampling 8x8 pada citra masukan.
* Menghitung µ dan σ Y, Cb, dan Cr dari masing-masing sampling.
* Menyimpan bobot training µ dan σ Y, Cb, dan Cr kedalam database.
* Load bobot training untuk menghitung matriks kovarian training sampel.
* Menghitung invers dari matriks kovarian.
* Menyimpan nilai invers dari matriks kovarian.

Proses pengujian meliputi:

* Menentukan jumlah K (K=1,3,5,7,9,11,13,15, 17 dan 19) yang akan digunakan.
* Masukkan data citra yang akan diuji.
* Mengonversi nilai RGB pada citra ke nilai YcbCr.
* Melakukan windowing citra 7x7.
* Menghitung µ dan σ Y, Cb, dan Cr dari masing-masing window.
* Load nilai invers matriks kovarian dari data latih untuk menghitung mahalanobis *distance*.
* Mengambil K distance terkecil untuk mengklasifikasi citra.
* Mendapatkan hasil klasifikasi warna kulit dari citra.

**3.2.4 Proses Pelatihan K-NN**

Diagram alir proses pelatihan K-NN dari penelitian ini adalah sebagai berikut :



Gambar 3.3 Proses Pelatihan K-NN

Dari Gambar 3.3 dapat dilihat bahwa dalam proses pelatihan yaitu membaca citra latih 64x64 piksel. Masukan yang merupakan nilai RGB dikonversikan ke YCbCr. Melakukan sampling citra 8x8 piksel dan dihtung µ dan σ dari Y, Cb dan Cr dari masing-masing bagian. Data µ dan σ dari Y, Cb dan Cr yang sudah didapat, disimpan kedalam *database*. Memanggil nilai µ dan σ dari Y, Cb dan Cr dari *database* untuk menghitung matriks kovarian dari 5 warna kulit dan bukan kulit. Menghitung invers matriks kovarian dari 5 warna kulit dan bukan kulit. Nilai invers yang didapatkan akan disimpan kedalam *file* txt.

**3.2.5 Diagram Alir Pengujian Citra**

Diagram Alir Proses Pengujian Citra dari Penelitian Ini adalah sebagai berikut :



Gambar 3.4 Diagram Alir Pengujian Citra

Dari Gambar 3.4 dapat dilihat bahwa dalam proses pengujian citra, sebelum citra diuji terlebih dahulu kita tentukan jumlah K yang akan digunakan. citra input yang akan diuji terlebih dahulu dikonversi dari ruang warna RGB ke ruang warna YCbCr, lakukan proses *windowing* pada matriks citra, dalam hal ini penulis menggunakan metode *windowing* matriks 7x7 untuk melakukan pengecekan matriks gambar per piksel. Setelah dilakukan *windowing*, hitung nilai µ dan σ Y,Cb, dan Cr dari masing-masing window. Nilai tersebut akan digunakan untuk menghitung *distance* pada *mahalanobis distance*. Setelah *distance*-nya didapatkan, maka diambil sejumlah K-*distance* terkecil. Sejumlah K-*distance* tersebut akan menentukan warna kulit atau bukan kulit dari citra uji.

**3.2.6 Proses Windowing Citra**

Pada proses pengenalan citra kulit dan bukan kulit ini dilakukan proses *filtering* citra dengan menggunakan filter *windowing*, filter *windowing* ini digunakan untuk mengecek masing-masing piksel dari citra yang dimasukkan. Adapun contoh cara kerja filter *windowing* adalah sebagai berikut :

|  |
| --- |
| Int [][]A={{5,4,7,6,5,8,6,7},{6,6,4,7,8,9,0,7},{8,6,7,1,2,5,6,10},{45,66,2,3,55,4,2,3},  {11,21,44,51,5,62,44,8},{5,6,3,7,8,2,4,11},{44,1,5,56,18,99,70,54},{8,5,3,9,6,4,3,3}};  //windowing 7x7 dari array 2 dimensi    6 4 7 8 9 0 7  6 7 1 2 5 6 10  66 2 3 55 4 2 3  21 44 51 5 62 44 8  6 3 7 8 2 4 11  1 5 56 18 99 70 54  5 3 9 6 4 3 3  6 6 4 7 8 9 0  8 6 7 1 2 5 6  45 66 2 3 55 4 2  11 21 44 51 5 62 44  5 6 3 7 8 2 4  44 1 5 56 18 99 70  8 5 3 9 6 4 3  5 4 7 6 5 8 6  6 6 4 7 8 9 0  8 6 7 1 2 5 6  45 66 2 3 55 4 2  11 21 44 51 5 62 44  5 6 3 7 8 2 4  44 1 5 56 18 99 70  4 7 6 5 8 6 7  6 4 7 8 9 0 7  6 7 1 2 5 6 10  66 2 3 55 4 2 3  21 44 51 5 62 44 8  6 3 7 8 2 4 11  1 5 56 18 99 70 54 |

Gambar 3.5 Contoh windowing

Pada Gambar dapat dilihat proses *windowing* matriks, ukuran *windowing* yang digunakan yaitu 7x7. Proses *windowing* 7x7 yang tampak pada contoh diatas dicari nilai *mean* (rata-rata) dan standar deviasi dari masing-masing matriksnya terlebih dahulu. Nilai *mean* dan standar deviasi yang didapatkan dari matriks-matriks tersebut kemudian akan dijadikan sebagai nilai masukan untuk perhitungan KNN.

**3.2.7 Parameter Mean dan Standar Deviasi**

Pada penelitian ini, digunakan nilai µ (mean) dan σ (Standar Deviasi) untuk menentukan ciri dari warna kulit pada citra. Alasan pemilihan µ (*mean*) dan σ (Standar Deviasi) sebagai parameter ciri yang digunakan pada penelitian ini, karena *mean* (rata-rata) merupakan suatu ukuran pemusatan data. *Mean* suatu data juga merupakan statistik karena mampu menggambarkan bahwa data berada pada kisaran rata-rata data tersebut (Susanti, 2014).

(3-1)

Dengan :

µ : Mean

: Data piksel ke-i,j

n : Jumlah data pada baris

m : Jumlah data pada kolom

Sedangkan Standar Deviasi merupakan Salah satu teknik statistik untuk menjelaskan homogenitas kelompok. Standar Deviasi merupakan variasi sebaran data. Semakin kecil nilai sebarannya berarti variasi nilai data makin sama Jika sebarannya bernilai 0, maka nilai semua datanya adalah sama. Semakin besar nilai sebarannya berarti data semakin bervariasi (Susanti, 2014).

(3-2)

Dengan :

σ : Standar deviasi

: Data ke-i

: Rata-rata X

n : Jumlah data

Sehingga dari penjelasan tentang µ (mean) dan σ (standar deviasi) diatas, dapat disimpulkan bahwa alasan penggunaan parameter µ (mean) dan σ (standar deviasi) sebagai ciri pada suatu citra yaitu karena mean dan standar deviasi pada masing-masing data citra akan memiliki nilai yang berbeda yang mewakili citra itu sendiri.

**3.2.8 Pengumpulan dan Pengujian Hasil**

Data citra dengan objek manusia dari berbagai warna kulit (akan diklasifikasikan ke dalam lima warna kulit yaitu hitam pekat, hitam, coklat, kuning langsat, dan putih), data citra dengan berbagai resolusi gambar, dan data citra dengan berbagai kecerahan yang dimiliki.

Pengujian yang akan dilakukan berfungsi untuk menentukan nilai efisiensi dari program yang akan dibuat berdasarkan algoritma K-NN (K-*Nearest Neighbor*) untuk menentukan warna kulit yang menggunakan pembagian kelas berdasarkan algoritma K-NN.

**BAB IV**

**HASIL DAN PEMBAHASAN**

Berdasarkan metodologi pada bab sebelumnya, maka pada bab ini dilakukan pemisahan warna kulit hitam pekat, hitam, coklat, kuning langsat, dan putih, serta bukan kulit. pada citra masukan, pengklasifikasian warna kulit akan dilakukan dengan menggunakan pengolahan citra dengan algoritma K-Nearest Neighbour (KNN). Algoritma ini merupakan algoritma yang yang sederhana, yang bekerja berdasarkan jarak terpendek dari data uji ke data yang telah dilatih.

* 1. **Persiapan Data Latih dan Data Uji**

Pada penelitian ini, data yang akan digunakan sebagai data latih dan data uji adalah data citra yang berfomat jpeg, yang diperoleh dari internet. Digunakan 600 sampel citra. 300 sebagai citra pelatihan dan 300 sebagai citra uji. Contoh gambar citra cari masing-masing kelas disajikan pada gambar berikut ini.







Gambar 4.1 Contoh Sampel Bukan Kulit







Gambar 4.2 Contoh Sampel Warna Kulit Hitam Pekat







Gambar 4.3 Contoh Sampel Warna Kulit Hitam







Gambar 4.4 Contoh Sampel Warna Kulit Coklat







Gambar 4.5 Contoh Sampel Warna Kulit Kuning Langsat





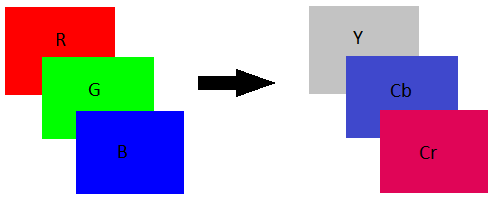


Gambar 4.6 Contoh Sampel Warna Kulit Putih

* 1. **Konversi Warna RGB ke YCbCr**

Citra berwarna dalam ruang warna RGB tidak sesuai jika digunakan untuk melakukan karakterisasi warna kulit. Dalam ruang RGB, terdapat tiga buah komponen (merah-hijau-biru) yang tidak hanya menyajikan warna saja, tetapi juga luminansi atau kecerahan. Luminansi antara wajah tiap orang berbeda-beda karena perbedaan kondisi pencahayaan dan ini tidak sesuai jika digunakan untuk memisahkan daerah kulit dari daerah yang bukan kulit. Luminansi dapat dihilangkan dari tampilan warna di dalam ruang warna kromatik. Warna kromatik dikenal juga dengan sebagai warna asli tanpa adanya luminansi, yang dapat dilakukan dengan proses normalisasi (Prasetyo, E.,2011).

Keuntungan utama dari mengubah citra ke ruang YCbCr ialah pengaruh luminansi dapat dihilangkan selama pemrosesan citra. Dalam ruang RGB, tiap komponen citra (merah, hijau, dan biru) mempunyai tingkat kecerahan yang berbeda-beda. Dengan demikian di dalam ruang YCbCr semua informasi tentang tingkat kecerahan diberikan oleh komponen Y, karena komponen Cb (biru) dan komponen Cr (merah) tidak tergantung dari luminansi.



Gambar 4.7 Konversi RGB ke YCbCr

Transformasi ruang warna RGB ke YCbCr berada pada persamaan (2-3).

Gambar 4.8 berikut merupakan contoh fungsi konversi ruang warna RGB ke YCbCr:

|  |
| --- |
| public int y2 (int red,int green,int blue){  y = (int) ( 0.299 \* red + 0.587 \* green + 0.114 \* blue);  return y;  }  public int cb2 (int red,int green,int blue) {  cb = (int)(-0.16874 \* red - 0.33126 \* green + 0.50000 \* blue);  return cb;  }  public int cr2 (int red,int green,int blue) {  cr = (int)( 0.50000 \* red- 0.41869 \* green- 0.08131 \* blue);  return cr;  } |

Gambar 4.8 Fungsi Konversi RGB ke YCbCr

* 1. **Proses KNN (K-Nearest Neighbour)**

Proses terakhir adalah klasifikasi menggunakan *k-nearest neighbour*. Proses ini dilakukan untuk mencari jarak terdekat antara data uji dan data latih, untuk mengetahui data uji tersebut masuk ke kelas yang mana. Dengan rumus *mahalanobis distance* yang berada pada persamaan (2-12), dihitung jarak terdekat antara data uji dengan data latih yang telah disimpan.

Parameter jarak yang digunakan adalah Mahalanobis dengan nilai K=1, K=3, K=5, K=7, K=9, K=11, K=13, K=15, K=17, K=19.

Sebelum melakukan perhitungan mahalanobis *distance* proses pertama yang dilakukan ialah mencari nilai matriks kovarian dari µY, µCb, µCr, σY, σCb , σCr masing-masing data latih citra dari setiap warna kulit. dari data tersebut akan menghasilkkan matriks kovarian berukuran 6x6 pada setiap warna kulit. Adapun rumus matriks kovarian yang digunakan berapa pada persamaan (2-13).

Gambar 4.9 berikut merupakan contoh perhitngan matriks kovarian.

|  |
| --- |
| public double[][] hitungcovarian(String kulit){  double matrikshasilkov[][]=new double[6][6];  double matrikscovarian[][]=new double [matrikshasilkov.length] [matrikshasilkov.length];  double [][]matrikskov=new double[temp.length][temp.length];  //penjumlahan matriks  for(int i=0;i<matrikskov.length;i++){  for(int j=0;j<matrikskov[i].length;j++){  matrikshasilkov[i][j]=matrikskov[i][j]+matrikshasilkov[i][j]; } }  for(int i=0;i<matrikshasilkov.length;i++){  for(int j=0;j<matrikshasilkov[i].length;j++){  matrikscovarian[i][j]= (matrikshasilkov[i][j]/(a.size()-1));  }  }  } |

Gambar 4.9 Contoh Fungsi Matriks Kovarian

Setelah matriks kovarian didapatkan, ,maka proses selanjutnya yaitu menghitung invers dari matriks yang telah didapatkan. Pada permasalahan ini, kami menggunakan perhitungan invers matriks dengan menggunakan eliminasi Gauss Jordan.

Berikut merupakn persamaan untuk mencari invers matriks dengan eliminasi Gauss-Jordan. (A|*I*)=(*I*|A-1). A merupakan matriks, *I* merupakan matriks identitas dan A-1 merupakan invers matriks A

Berikut contoh penggunaan eliminasi Gauss Jordan. Pada contoh digunakan matriks 3x3.

Selanjutnya masukkan matriks kedalam elimminasi Gauss-jordan. Bentuknya seperti berikut :

Untuk mencari invers matriks, gimna caranya agar matriks identitas berada disebelah kiri. Baris berapa saja bisa dikali, dijumlah, dibagi atau dikurangi.

Kurangi baris ketiga dengan 5x baris pertama :

Tambahkan baris ketiga dengan 4x baris kedua dari matriks diatas :

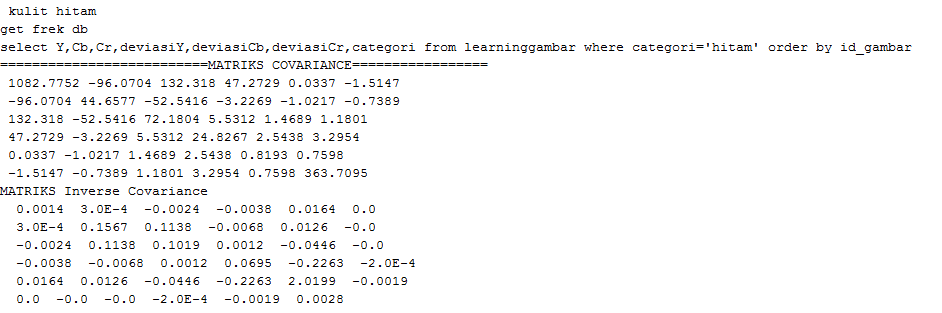
Kurangkan baris kedua dengan 4x baris pertama. Kurangkan baris pertama dengan 3x baris ketiga :

Kurangkan baris pertama dengan 2x baris kedua :

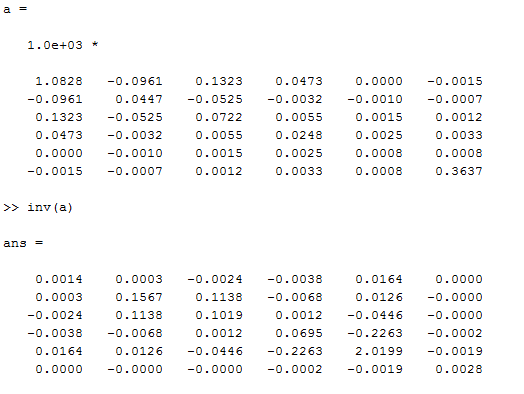
Setelah matriks identitas berada diseblah kiri, maka selesai sudah proses yang dilakukan. Dan matriks yang berada disebalh kanan merupakan invers dari matriks A.

* 1. **Pengujian Sistem**

Program yang telah dibuat, akan diuji dengan *mathlab* agar dapat dipastikan bahwa sistem berjalan sesuai dengan yang diinginkan.



Gambar 4.10 Contoh Hasil Output pada Sistem



Gambar 4.11 Contoh Hasil Output dari *Mathlab*

Sistem yang telah dibangun dengan menggunakan aplikasi *JAVA* telah diuji dengan menggunakan aplikasi *mathlab*. Dari gambar 4.10 dan 4.11 dapat dilihat bahwa sistem yang di bangun sudah sesuai seperti yang diharapkan. Bagian dari sistem yang di uji yaitu pada perhitungan invers matriks kovarian dari warna kulit hitam pekat, hitam, coklat, kuning langsat dan putih serta matriks kovarian dari bukan kulit.

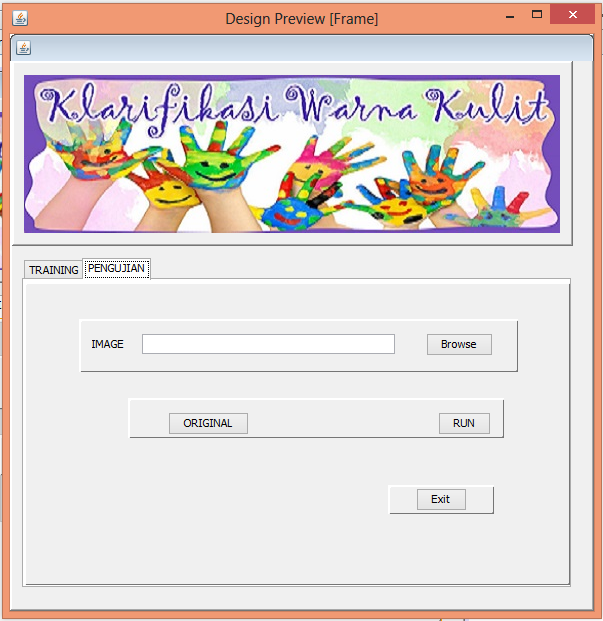
* 1. **Antarmuka**

Setelah program dijalankan, adapun tampilan antarmukanya sebagai berikut :



Gambar 4.12 Tampilan Proses Pelatihan

Dari Gambar 4.12 dapat dilihat bahwa langkah pertama pada proses KNN yaitu proses pelatihan. Masukkan *path* dari data latih dengan menekan tombol *browse* dan memilih kelas yang akan dilatih. Terdapat 6 kelas yang akan dilatih, yaitu bukan kulit (*non*), hitam pekat, hitam, coklat, kuning langsat, dan putih. Pelatihan akan dilaksanakan setelah menekan tombol *RUN.* Data yang telah dilatih akan di *reset* apabila menekan tombol *RESET.* Untuk menampilkan nilai invers dari pelatihan maka tekan tombol DATA INVERS. Serta tombol *EXIT* berfungsi untuk keluar dari program.

Gambar 4.13 Tampilan Proses Pengujian

Langkah selanjutnya adalah proses untuk melakukan pengujian seperti gambar 4.13. Pertama masukkan citra uji dengan menekan tombol *Browse*. Tombol *original* akan menampilkan citra uji. Kemudian dilakukan proses pengujian pada tombol *RUN*, yang akan menampilkan citra uji dan hasil dari pengujian berupa penentuan warna kulit. Waktu komputasi untuk proses pengujian dari citra di atas adalah 86487 *milisecond*.

* 1. **Hasil Pengujian**

Hasil analisisi *confusion matrix* dengan data latih 100 perkelas. Semua data latih di acak 50 sebagai data latih dan 50 sebagai data uji yang dilakukan disetiap kelas.

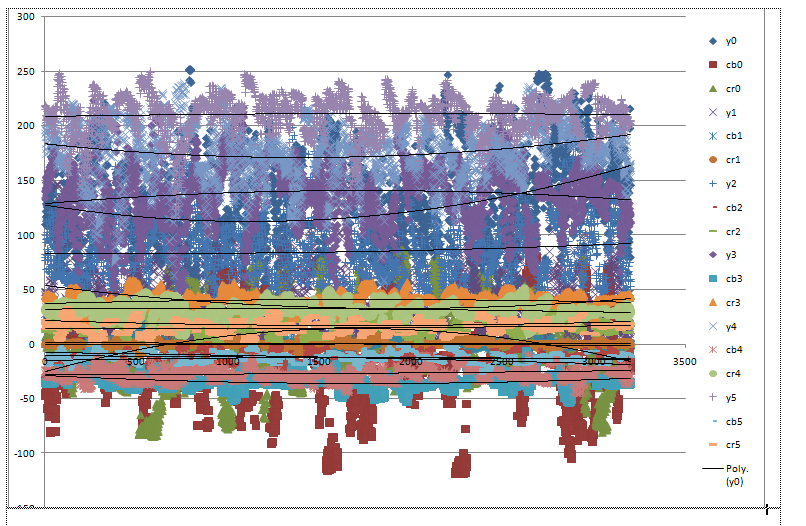
Berdasarkan persamaan (2-9), (2-10) dan (2-11) maka dihasilkan grafik seperti berikut ini.

Berikut merupakan grafik akurasi, FPR dan FNR dari 10 variasi K :

Gambar 4.14. Grafik ROC dari Pengujian Citra dengan 10 Variasi K

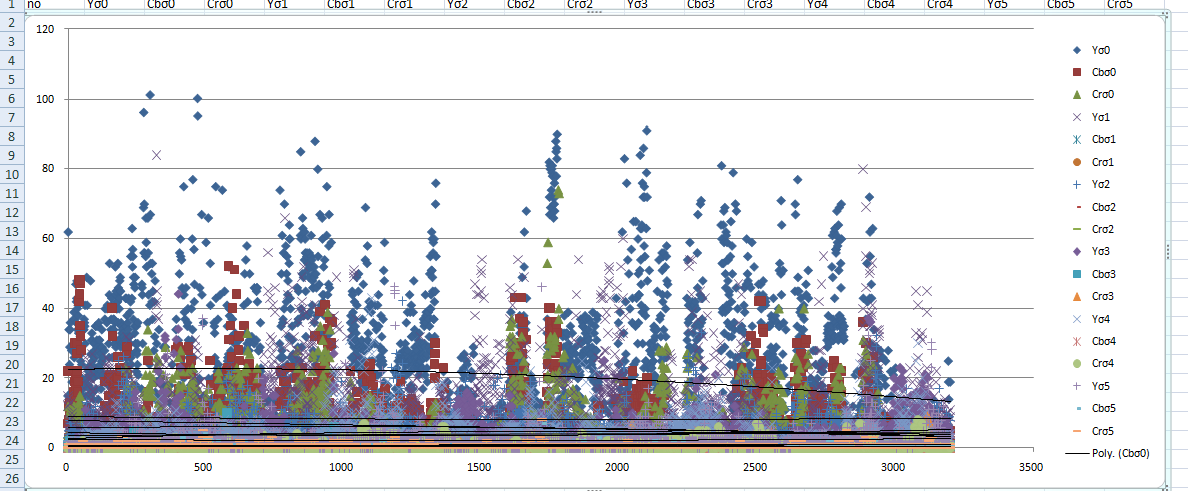
Dari gambar 4.14 dapat dilihat hasil terbaik jatuh pada K=13 dengan nilai akurasi sebesar 78%, dengan nilai FPR 19.12% serta FNR 24.38. Nilai akurasi yang tidak terlalu tinggi disebabkan oleh penyebaran data dari satu kelas, memiliki data yang beririsan dengan data dari kelas lainnya.

Berikut merupakan grafik penyebaran data dari Y, Cb, dan Cr dari masing-masing kelas :



Gambar 4.15 Penyebaran Data Mean Dari Y, Cb, Dan Cr Masing-Masing Kelas

Berikut merupakan penyebaran data standar deviasi dari Y, Cb, dan Cr dari masing-masing kelas :



Gambar 4.16 Penyebaran Data Standar Deviasi Dari Y, Cb Dan Cr Masing-Masing Kelas

Keterangan dari gambar 4.15 dan 4.16 :

Y0, Cb0, Cr0 : data Cb dan Cr dari bukan kulit

Y1, Cb1, Cr1 : data Cb dan Cr dari kelas hitam pekat

Y2, Cb2, Cr2 : data Cb dan Cr dari kelas hitam

Y3, Cb3, Cr3 : data Cb dan Cr dari kelas coklat

Y4, Cb4, Cr4 : data Cb dan Cr dari kelas kuning langsat

Y5, Cb5, Cr5 : data Cb dan Cr dari kelas putih

Dari gambar 4.15 dan 4.16 dapat dilihat penyebaran data dari masing-masing warna kulit dan bukan kulit. terjadi pengelompokan nilai dari setiap kelasnya dan ada nilai dari setiap kelas yang beririsan dengan kelas yang lain. Data diambil dari nilai µ dan σ data latih dari masing-masing kelas.

Hasil perhitungan *Confusion Matrix* dari tiap kelas dengan parameter K=13 adalah seperti pada tabel berikut:

Tabel 4.1 Hasil Kalkulasi dari K 13

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kelas | TP | FP | TN | FN | Akurasi (%) |
| Bukan kulit | 50 | 0 | 34 | 16 | 84 |
| Hitam Pekat | 39 | 11 | 39 | 11 | 78 |
| Hitam | 42 | 8 | 38 | 12 | 80 |
| Coklat | 40 | 10 | 35 | 15 | 75 |
| Kuning langsat | 30 | 20 | 39 | 11 | 69 |
| Putih | 47 | 3 | 35 | 15 | 82 |
| Rata-rata | 41.33 | 8.67 | 36.67 | 13.33 | 78 |

Dari tabel 4.1 hasil kalkulasi dengan K=13 diatas, dapat dilihat bahwa akurasi terbaik terdapat pada kelas bukankulit dengan akurasi sebesar 84%. Berdasarkan tabel, dapat dilihat grafik akurasi, FpR dan FnR dari K 13 pada gambar 4.17.

Gambar 4.17 Grafik Akurasi, FPR, dan FNR pada K=13

Keterangan kelas untuk gambar 4.17 adalah :

1. Bukan kulit
2. Hitam pekat
3. Hitam
4. Coklat
5. Putih
6. Kuning langsat

Dari gambar 4.17 dapat dilihat bahwa akurasi terbesar pada kelas bukan kulit yaitu sebesar 84%. FPR terkecil pada kelas bukan kulit dengan persentase sebanyak 0%. Sedangkan FNR terkecil pada kelas putih dengan persentase 7.89%.

* 1. **Waktu Komputasi**

Berikut merupakan grafik dari waktu komputasi dari beberapa kombinasi ukuran citra yang digunakan.

Gambar 4.18 Grafik Waktu Komputasi Citra

Keterangan ukuran citra pada gambar 4.18 yaitu :

1. Citra ukuran 64x64 piksel
2. Citra ukuran 100x100 piksel
3. Citra ukuran 200x200 piksel
4. Citra ukuran 300x300 piksel
5. Citra ukuran 400x400 piksel

Dari gambar 4.18 dapat dilihat bahwa waktu komputasi paling kecil yaitu citra ukuran 64x64 piksel dengan waktu komputasi 24.29 *second*. Sedangkan waktu komputasi paling lama yaitu pada citra ukuran 400x400 piksel dengan waktu yang dibutuhkan 100.73 *second*. Ukuran citra dapat mempengaruhi waktu komputiasi dikarenakan semakin besar citra uji, semakin banyak perhitungan yang terjadi sehingga menyebabkan semakin lama waktu komputasi.

* 1. **Contoh Klasifikasi Kulit dan Bukan Kulit**

Berikut merupakan beberapa contoh dari klasifikasi warna kulit yang telah dilakukan pada saat pengujian.

Tabel 4.2 Contoh Hasil Klasifikasi Warna Kulit yang Berhasil dan Gagal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Citra Uji | Warna Kulit Citra Uji | Hasil pengujian | keterangan |
| 1 | 2 | 3 | 4 | 5 |
| 1 | C:\DataLatih\0\18.jpg | Bukan kulit | Bukan kulit | Berhasil |
| 2 | C:\DataLatih\1\18.jpg | Hitam pekat | Hitam pekat | Berhasil |
| 3 | C:\DataLatih\2\29.jpg | Hitam | Hitam | Berhasil |
| 4 | C:\DataLatih\3\54.jpg | Coklat | Coklat | Berhasil |
| 5 | C:\DataLatih\4\23.jpg | Kuning langsat | Kuning langsat | Berhasil |
| 6 | C:\DataLatih\5\19.jpg | Putih | Putih | Berhasil |
| 7 | C:\DataLatih\1\90.jpg | Hitam pekat | Bukan kulit | Gagal |
| 1 | 2 | 3 | 4 | 5 |
| 8 | C:\DataLatih\2\20.jpg | Hitam | Bukan kulit | Gagal |
| 9 | C:\DataLatih\3\35.jpg | Coklat | Kuning langsat | Gagal |
| 10 | C:\DataLatih\4\81.jpg | Kuning langsat | Putih | Gagal |
| 11 | C:\DataLatih\5\27.jpg | Putih | Bukan kulit | Gagal |

Dari tabel 4.2 dapat dilihat contoh dari hasil klasifikasi yang dinyatakan gagal dan hasil klasifikasi yang dinyatakan berhasil. Citra yang digunakan yaitu citra dari 5 warna kulit dan bukan kulit yang memiliki ukuran 64x64 piksel.

**4.9 Contoh Klasifikasi Warna Kulit dengan Kulit Hewan (Bukan Kulit)**

Berikut meerupakan hasil klasifikasi dari warna kulit dengan menggunakan kulit hewan :

Tabel 4.3 Contoh Hasil Klasifikasi dengan Kulit Hewan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Citra Uji | Nama Hewan | Hasil Pengujian | Keterangan |
|
| 1 | 2 | 3 | 4 | 5 |
|
| 1 | C:\Users\MATRI\Documents\0000 data hewan\kambinng.jpg | Kambing | Bukan Kuiit | Berhasil |
|
| 2 | C:\Users\MATRI\Documents\0000 data hewan\kerbau.jpg | Kerbau | Bukan Kulit | Berhasil |
|
| 1 | 2 | 3 | 4 | 5 |
| 3 | C:\Users\MATRI\Documents\0000 data hewan\kuda.jpg | Kuda | Hitam | Gagal |
| 4 | C:\Users\MATRI\Documents\0000 data hewan\orangutan.jpg | Orang Utan | Hitam | Gagal |
|
| 5 | C:\Users\MATRI\Documents\0000 data hewan\sapi bali.jpg | Sapi | Bukan Kulit | Berhasil |
|

Dari tabel 4.3 dapat dilihat hasil dari kasifikasi warna kulit dengan menggunakan kulit hewan. Pada klasifikasi ini, kulit hewan yang digunakan yaitu Kambing, Kerbau, Kuda, Orang utan, dan sapi. Dari kelima contoh tersebut, Kuda dan Orang utan dinyatakan gagal karena hasil pengujian dari hewan tersebut berwarna kulit hitam. Sedangkan Kerbau, Kambing, dan Sapi dinyatakan berhasil karena hasil pengujiannya tidak termasuk kedalam warna kulit.

**BAB V**

**PENUTUP**

* 1. **Kesimpulan**

Berdasarkan penelitian pada BAB sebelumnya, maka dapat disimpukan bahwa:

1. Penentuan warna kulit cukup efektif dengan menggunakan metode KNN dengan dengan akurasi sebesar 78% , dengan nilai FPR 19.12% serta FNR 24.38 pada K=13.
2. Hasil dari kalkulasi pada K=13, kelas kulit hitam pekat memiliki akurasi sebesar 78%, nilai FPR 22%, nilai FNR 22%. Pada kulit hitam memiliki akurasi sebesar 80%, nilai FPR 17.4%, dan nilai FNR 22.2%. Kulit coklat memiliki akurasi sebesar 75%, nilai FPR 22.2%, dan nilai FNR 27.3%. Kulit kuning langsat memiliki akurasi sebesar 69% , nilai FPR33.8% , dan nilai FNR 26.8%. Kulit putih memiliki akurasi 82%, nilai FPR 7.8%, dan nilai FNR 24.2% .
   1. **Saran**
3. Dalam melakukan proses pengkalsifikasian warna kulit menggunakan algoritma K-NN ini dibutuhkan spesifikasi komputer yang cukup baik, agar memperlancar proses komputasi.
4. Dalam melakukan pemilihan pada data pelatihan sebaiknya lebih bervariasi dan dilakukan pengecekan tiap kelasnya, agar di kelas yang 1 dengan kelas yang lain tidak memiliki kesamaan nilai.