

BAB II TINJAUAN PUSTAKA

2.1 Penelitian yang Terkait

Langkah awal teknik deteksi kulit manusia pada metode dengan pendekatan *pixel-based* secara umum adalah dengan melakukan transformasi dari *color space* citra asli menjadi *color space* lain yang dipilih. Kemudian klasifikasi dilakukan pada setiap *pixel* dalam citra dengan cara memberikan suatu model klasifikasi yang dapat berbentuk *rule* atau dapat juga berupa batasan rentang nilai *threshold/boundary* [4]. *Pixel* yang memenuhi kriteria klasifikasi model atau memiliki nilai komponen *color space* yang termasuk dalam rentang *threshold* akan diklasifikasikan sebagai kulit, sementara *pixel* yang berada di luar rentang *threshold* akan diklasifikasikan sebagai bukan kulit.

Beberapa *color space* yang digunakan di antaranya adalah RGB, yCbCr, HSV, CIE Lab, CIE Luv, dan *normalized* RGB. Vezhnevets [6], Phung [18], Kakumanu [1], dan Khan [19] juga telah melakukan perbandingan di antara model-model *color space* tersebut. RGB *color space* adalah *color space default* untuk kebanyakan representasi dan penyimpanan format citra, sedangkan *color space* yang lain didapatkan dengan cara melakukan transformasi baik linier maupun *non* linier dari nilai intensitas R, G, dan B dalam *color space* RGB.

Beberapa contoh nilai *threshold* statis yang digunakan pada metode deteksi kulit manusia dengan pendekatan *pixel-based* di antaranya adalah Sobottka dan Pitas [20] menggunakan rentang nilai *threshold* statis $RH = [0, 50]$ dan $RS = [0.23, 0.68]$ dalam *color space* HS untuk mendapatkan *pixel* dalam citra yang dianggap kulit manusia. Deteksi kulit manusia dalam *color space* yCbCr diusulkan oleh Chai & Ngan [21] menggunakan rentang nilai $Cb_{max} = 127$; $Cb_{min} = 77$; $Cr_{max} = 173$; $Cr_{min} = 133$. Wang dan Yuan [22] juga menggunakan rentang nilai *threshold* statis $R_r = [0.36, 0.465]$, $R_g = [0.28, 0.363]$, $RH = [0, 50]$, $RS = [0.20, 0.68]$, dan $RV = [0.35, 1.0]$ dalam *color space* HSV untuk membedakan mana *pixel* yang kulit dan mana *pixel* yang bukan kulit. Peer [23] mengusulkan metode pendeteksian kulit manusia dalam

color space RGB dengan *range threshold* yang juga statis yaitu $R > 95$; $G > 40$; $B > 20$; $(\text{Max}\{R, G, B\} - \text{min}\{R, G, B\}) > 15$; $|R-G| > 15$; $R > G$; $R > B$.

Permasalahan yang ada pada metode dengan pendekatan *pixel-based* adalah rendahnya kinerja dan ketahanan pada saat dihadapkan dengan banyaknya variasi warna kulit manusia karena perbedaan etnis, bermacam-macam karakteristik kamera, pengaruh kompleksitas *background* citra [1], dan sensitif terhadap perubahan iluminasi. Ketahanan terhadap perubahan iluminasi dapat lebih baik ketika menggunakan beberapa model *color space* dengan cara tidak melibatkan komponen *luminance* pada perhitungan [18]. Sebagai contoh dalam *color space* *yCbCr*, yang dipertimbangkan hanya komponen *Cb* dan *Cr* yang merupakan *chrominance*, sedangkan nilai komponen *y* yang merupakan komponen *luminance* diabaikan dan tidak dimasukkan dalam perhitungan.

Permasalahan tersebut timbul karena menurut Vezhnevets [6], model klasifikasi yang dibangun pada metode dengan pendekatan *pixel-based* bersifat statis dan dibuat dengan tujuan untuk dapat melakukan klasifikasi secara *general* agar lebih tahan terhadap berbagai macam variasi warna kulit manusia tersebut.

Yogarajah [7] mengusulkan penggunaan metode *online dynamic threshold* untuk mengatasi masalah tersebut. Metode ini menggunakan model *color space* yang diusulkan oleh Cheddad [24], yang merupakan modifikasi *color space* RGB dengan menggunakan *transformation matrix*.

Metode ini memanfaatkan informasi kulit pada region wajah untuk menentukan dan mencari *pixel* lain dalam citra yang diduga sebagai kulit. Kulit wajah digunakan dengan dasar pendapat yang dikemukakan Zheng [9] bahwa kulit wajah dan kulit tubuh manusia selalu memiliki distribusi warna yang sama. *Region of interest* (ROI) wajah ditemukan dengan terlebih dahulu mendeteksi posisi mata pada citra dengan algoritma pendeteksian mata menggunakan *Machine Perception Toolbox* (MPT) yang diusulkan oleh Fasel [8]. Dari posisi mata yang ditemukan tersebut maka dapat dihitung suatu *region* bentuk elips yang dianggap sebagai *region* wajah dalam citra. *Region* yang diduga sebagai wajah tersebut kemudian dihilangkan bagian *non-smoothing*-nya dengan menggunakan deteksi tepi *sobel* dan operasi dilasi. Dari ROI

wajah inilah maka akan dicari nilai *threshold* untuk mencari *pixel* kulit manusia lain dalam citra testing tersebut

Tan [10], mengusulkan perbaikan pada *online dynamic threshold* yang diusulkan oleh Yogarajah [7] dengan penambahan *feature* yaitu *fusion* antara *Gaussian model* dengan *2D-smoothing histogram*. *Color space* yang digunakan pada metode ini tidak lagi menggunakan *transformation matrix* dari *color space* RGB yang diusulkan Cheddad [24] tetapi menggunakan *log-opponent chromacity* (LO), karena LO *color space* dianggap memiliki relevansi dengan persepsi visual pada sistem penglihatan manusia.

Penggunaan *online dynamic threshold* yang dikembangkan Yogarajah [7] dan Tan [10] secara umum dapat mengatasi permasalahan yang dihadapi metode deteksi kulit manusia pada pendekatan *pixel-based*, karena metode ini mampu menghasilkan model klasifikasi yang bersifat dinamis. Model akan berubah dan beradaptasi sesuai dengan kondisi *dataset* testing, sehingga diharapkan tahan terhadap berbagai macam variasi warna kulit manusia.

Namun metode ini masih memiliki kelemahan yaitu sangat tergantung pada keberhasilan algoritma pendeteksian posisi kedua mata yang diusulkan oleh Fasel [8]. Ditemukannya posisi koordinat kedua mata dalam citra sangat penting pada metode ini karena posisi mata dipergunakan untuk menghitung posisi koordinat ROI wajah, sehingga jika kedua mata tidak berhasil dideteksi maka langkah berikutnya tidak dapat dikerjakan. Pada saat dihadapkan dengan citra yang berisi subjek wajah manusia dalam posisi yang relatif miring atau juga dalam posisi agak menyamping maka algoritma deteksi mata yang diusulkan oleh Fasel [8] ini akan gagal.

Kelemahan berikutnya dari metode *fusion* antara *Gaussian model* dan *2D-smoothing histogram* yang diusulkan Tan [10] adalah masih rendahnya kinerja yang dihasilkan metode ini. Saat diujikan dengan *dataset* Pratheepan Yogarajah [7], metode yang diusulkan Tan [10] mendapatkan akurasi 84,95%, *precision rate* 55,31%, dan *recall rate* sebesar 56,22%. Pada eksperimen pada *dataset* tersebut, luas *Area Under the ROC Curve* (AUC) yang didapatkan adalah 0,7350, sehingga menurut Gorunescu [11] metode yang diusulkan Tan [10] ini hanya dapat

digolongkan sebagai *fair classification*. Sedangkan nilai kappa yang didapatkan adalah 0,4669 sehingga menurut Landis [12] termasuk *moderate* dan menurut Fleis [13] digolongkan sebagai *fair*.

Kinerja yang kurang baik pada metode *fusion* antara *Gaussian model* dan *2D-smoothing histogram* yang diusulkan Tan [10] pada penelitian terdahulu ini, disebabkan oleh kelemahan saat pemisahan *smooth region* dengan *non-smooth region* pada ROI wajah yang terdeteksi. Pemisahan *smooth region* dengan *non-smooth region* dilakukan hanya dengan menggunakan operasi sederhana yaitu deteksi tepi *sobel* dan dilasi. Penyebab lain yang bisa diidentifikasi dari kurang baiknya kinerja metode ini adalah ketika beberapa ROI wajah terdeteksi dalam citra *multiple object*, maka setiap ROI wajah digunakan untuk mencari *pixel* lain dalam citra yang diduga sebagai kulit manusia, dan hasilnya langsung digabungkan dengan hasil dari ROI wajah yang lain.

Permasalahan lain pada metode *fusion* antara *Gaussian model* dan *2D-smoothing histogram* yang diusulkan Tan [10] adalah waktu pemrosesan yang relatif lama. Pada pengujian 78 citra pada *dataset* Pratheepan Yogarajah [7], rata-rata waktu untuk menghasilkan *output* deteksi kulit manusia pada 1 buah citra mencapai 21,25 detik.

2.2 Landasan Teori

2.2.1 HSV Color Space

Color space HSV termasuk *perceptual color space* seperti juga HSI dan HSL. *Color space* HSV terdiri dari komponen H (*hue*), S (*saturation*), dan V (*value*). Nilai-nilai komponen H, S, dan V dapat diperoleh dengan cara konversi dari komponen R (*red*), G (*green*), dan B (*blue*) pada *color space* RGB menggunakan transformasi *non-linear* berikut:

$$HSV : \begin{cases} H = \arccos \frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{((R-G)^2+(R-B)(G-B))}} \\ S = 1 - 3 \frac{\min(R, G, B)}{R + G + B} \\ V = \frac{1}{3}(R + G + B) \end{cases} \quad (1)$$

2.2.2 yCbCr Color Space

Color space yCbCr terdiri dari 3 komponen yaitu y, Cb, dan Cr. Komponen y adalah *luminance*, Cb adalah *chrominance blue*, dan Cr adalah *chrominance red*. *Color space* ini dirancang sebagai *color space* yang digunakan pada transmisi siaran televisi. Penggunaan lain *color space* yCbCr ini banyak ditemukan pada sistem *video coding* dan *video compression* seperti contohnya MPEG.

Pada kebanyakan format penyimpanan dan representasi citra, *color space* RGB adalah *color space default*. *Color space* yang lain didapatkan dengan cara melakukan transformasi baik linier maupun *non* linier dari nilai intensitas R,G, dan B dalam *color space* RGB. *Color space* yCbCr adalah salah satu *color space* yang ditransformasikan secara linear dari *color space* RGB dengan persamaan berikut:

$$yCbCr : \begin{cases} y = 0,299R + 0,587G + 0,114B \\ Cb = 0,56 (B - Y) \\ Cr = 0,71 (R - Y) \end{cases} \quad (2)$$

Color space yCbCr merepresentasikan secara independen komponen *luminance* (y) dari komponen *chrominance* (Cb dan Cr), sehingga dengan mudah dapat dilakukan pemisahan secara langsung antara komponen *luminance* dengan *chrominance* pada citra.

Khan [19] menyebutkan bahwa ketahanan terhadap perubahan iluminasi dapat lebih baik pada beberapa model *color space* dengan cara tidak melibatkan komponen *luminance* [6], sehingga untuk *color space* yCbCr maka hanya digunakan komponen Cb dan Cr yang merupakan komponen *chrominance*.

2.2.3 Deteksi Kulit Manusia dengan Pendekatan *Pixel-Based*

Vezhnevets [6] dan Kakumanu [1] menggolongkan beberapa model deteksi kulit manusia yang digunakan pada metode dengan pendekatan *pixel-based* sebagai berikut:

1. *Explicitly defined skin region*

Teknik *explicitly defined skin region* yang disebut juga *fixed decision boundary* atau *fixed thresholding* menggunakan satu atau beberapa batasan (*boundary*) rentang nilai *threshold* tertentu pada komponen-komponen *color space*, untuk mengambil keputusan (*decision*) *pixel* mana saja yang diklasifikasikan sebagai kulit. *Pixel* yang memiliki nilai komponen *color space* yang termasuk dalam rentang *threshold* akan diklasifikasikan sebagai kulit dan jika tidak maka akan diklasifikasikan sebagai bukan kulit.

2. *Non-parametric skin distribution modeling*

Pada teknik *non-parametric* ini *pixel* dalam *color space* direpresentasikan dalam bentuk *skin probability map* (SPM).

3. *Parametric skin distribution modeling*

Pada teknik *parametric*, *pixel* pada *color space* yang dipilih direpresentasikan ke dalam bentuk *histogram*, kemudian *histogram* distribusi warna tersebut dikonversikan menjadi bentuk *probability density function* (PDF). Pada kurva PDF ini maka *pixel* yang memiliki frekuensi kemunculan (probabilitas) lebih tinggi dari *threshold* yang diberikan akan diklasifikasikan sebagai kulit. Metode yang disulkan pada penelitian ini termasuk dalam jenis *paramateric skin distribution modeling*.

4. *Neural network*

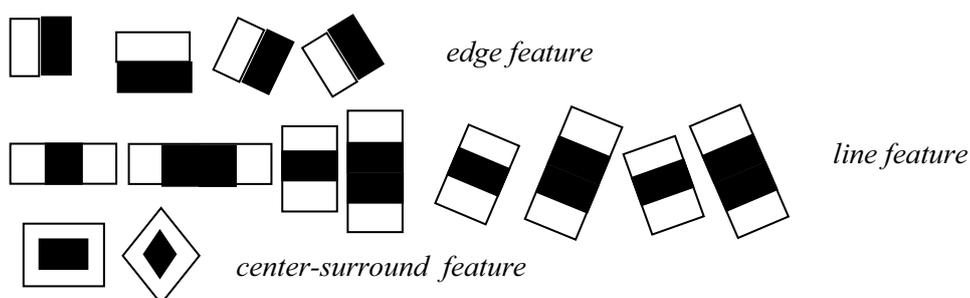
Pada teknik *neural network*, dilakukan *training* pada 2 jenis *database* yaitu *database positif* dan *database negatif* untuk mendapatkan model klasifikasi. Arsitektur *neural network* yang biasa digunakan adalah *multi-layer perceptron* (MLP) dan *self organized map* (SOM).

2.2.4 Object Detection Framework Viola-Jones

Penerapan *object detection framework* menggunakan yang algoritma *adaboost* ini pertama kali diperkenalkan oleh Paul Viola dan Michael Jones [14]. *Object detection framework* Viola-Jones menggunakan teknik menghitung *feature-feature* citra dengan efisien menggunakan *integral image*. *Feature-feature* yang telah dihitung dengan *integral image* akan membentuk beberapa *classifier* lemah. *Adaboost* akan memilih dan melakukan seleksi pada *classifier-classifier* lemah tersebut untuk membentuk *classifier kuat*. Selanjutnya dibangun *cascade detector* dengan tujuan meningkatkan kinerja deteksi, dan secara drastis akan mengurangi waktu komputasi.

2.2.4.1 Haar Feature dan Integral Image

Operator Haar digunakan untuk melakukan ekstraksi *feature*. Operator ini sederhana dan tahan terhadap perubahan iluminasi, sehingga *haar feature* dipilih sebagai *classifier* lemah. Gambar 1 menunjukkan *feature* persegi panjang *haar feature*. Sebuah persegi panjang *haar feature* dapat didefinisikan sebagai selisih dari jumlah *pixel* dari daerah di dalam persegi panjang, yang ada di setiap posisi dan skala dalam citra asli.



Gambar 1 Haar feature

Feature persegi panjang dapat dihitung dengan sangat cepat menggunakan representasi *intermediate* untuk citra yang disebut *integral image*. *Integral image* di lokasi x, y berisi jumlah *pixel* atas dan ke kiri dari x, y , yang direpresentasikan sebagai:

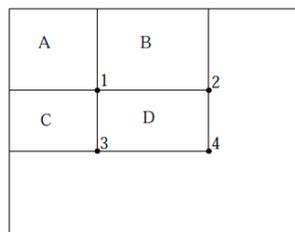
$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (3)$$

Di mana $ii(x, y)$ adalah *integral image* dan $i(x, y)$ adalah citra asli.

$$s(x, y) = s(s, y - 1) + i(x, y) \quad (4)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (5)$$

Dengan menggunakan *integral image*, maka setiap jumlah persegi panjang dapat dihitung dalam empat referensi *array*. Untuk lebih jelasnya bisa dilihat pada gambar 2. Dapat dilihat bahwa perbedaan antara dua jumlah persegi panjang dapat dihitung dalam delapan referensi.



Gambar 2 *Integral image*

Nilai *integral image* pada lokasi 1 adalah jumlah *pixel* adalah jumlah persegi panjang pada persegi panjang A. Nilai pada lokasi 2 adalah A+B. Nilai pada lokasi 3 adalah A+C. Jumlah nilai D dapat dihitung sebagai $4+1-(2-3)$.

2.2.4.2 *Adaptive Boosting (Adaboost)*

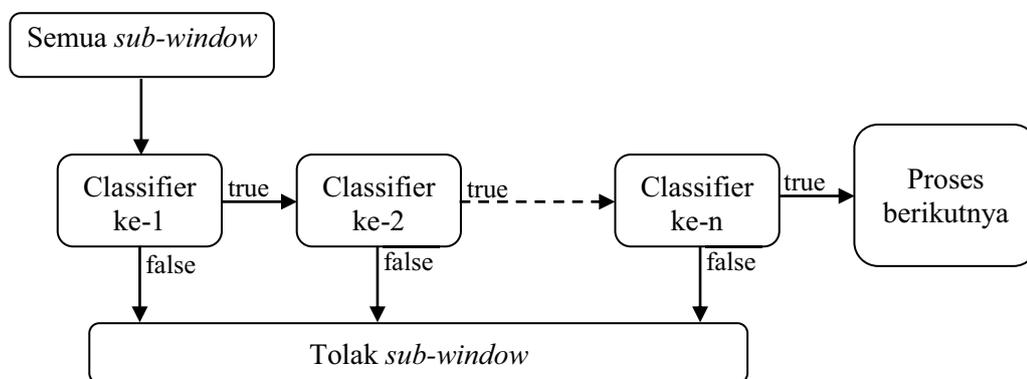
Algoritma *boost* adalah jenis algoritma yang universal, bertujuan untuk meningkatkan kinerja algoritma *learning*. Sebuah algoritma *boost* diterapkan untuk menemukan sejumlah *classifier* lemah melalui algoritma iteratif, dan mengintegrasikan *classifier* lemah ini menjadi *classifier* kuat dengan mempertimbangkan masing-masing bobot *classifier*. *Classifier* kuat ini kemudian diterapkan pada *subwindow* citra untuk mencari dimana kemungkinan lokasi objek yang sedang dicari.

Algoritma *boost* kemudian berkembang menjadi *adaptive boosting* (*Adaboost*). *Adaboost* pertama kali diusulkan oleh Freund dan Schapire [25]. Algoritma ini secara otomatis dapat memilih beberapa *classifier* lemah dari beberapa *classifier* lemah yang ada untuk membangun sebuah *classifier* yang kuat melalui integrasi terbobot *classifier* lemah yang terpilih. Algoritma ini disebut *Discrete AdaBoost* (DAB). DAB berkembang menjadi *Real Adaboost* (RAB) yang memiliki konvergensi yang lebih cepat. Friedman [26] mengusulkan suatu algoritma yang disebut sebagai *Gentle AdaBoost* (GAB) untuk mengatasi masalah kestabilan komputasi numerik dari algoritma RAB.

2.2.4.3 Cascade Structure Detector

Classifier cascade detector meningkatkan kinerja deteksi dan secara drastis mengurangi waktu komputasi. *Classifier* berbasis *boosting* ini mampu menolak banyak *subwindow* negatif dan dapat mendeteksi hampir semua kasus positif dengan menentukan *threshold* sebuah *classifier* berbasis *boosting* sedemikian sehingga tingkat *false negative* mendekati nol. *Classifier* sederhana berada di sebelum *classifier* lebih kompleks agar tingkat *false positive* yang rendah dapat tercapai.

Hasil positif dari *classifier* pertama diteruskan kepada *classifier* kedua yang juga telah disesuaikan untuk mencapai tingkat deteksi yang sangat tinggi. Hasil positif dari kedua *classifier* kemudian diteruskan kepada *classifier* ketiga, dan seterusnya. Sebuah hasil negatif pada setiap titik mengarah pada penolakan langsung dari *subwindow*.



Gambar 3 Cascade structure

Tahapan dalam *cascade* ini dibangun oleh pelatihan *classifier* menggunakan *adaboost* dan kemudian menyesuaikan nilai *threshold* untuk meminimalkan *false negatif*.

Proses pelatihan *cascade* dengan lebih banyak *feature* akan mencapai tingkat deteksi lebih tinggi dan *false positive* yang lebih rendah. Namun *classifier* dengan *feature* yang lebih banyak juga akan membutuhkan lebih banyak waktu untuk melakukan perhitungan. Optimalisasi dapat dilakukan dengan mengatur jumlah tahap *classifier*, jumlah *feature* di setiap tahap, dan nilai *threshold* setiap tahap.

2.2.5 Normal Probability Density Function

Distribusi probabilitas yang terpenting dalam statistik adalah distribusi normal atau disebut juga distribusi Gaussian. Montgomery [27] menyebutkan fungsi *probability density function* pada setiap *pixel* x dengan *mean* μ dan *variance* σ^2 yang memiliki distribusi normal adalah:

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (6)$$

dimana $\pi = pi$, konstanta yang nilainya sama dengan 3.1416

$e = exponen$, konstanta yang nilainya 2,7183

$\mu =$ nilai rata-rata (*mean*) dari distribusi populasi

$\sigma =$ standar deviasi (simpangan baku) dari distribusi populasi

$\sigma^2 = variance$

Sifat distribusi normal:

1. Grafiknya selalu terletak di atas sumbu x .
2. Rata-ratanya (*mean*) μ dan standar deviasinya $= \sigma$.
3. Titik belok tepat di $x=\mu\pm\sigma$.
4. Mode (*maximum*) terjadi di $x=\mu$.
5. Bentuk grafiknya simetris terhadap $x = \mu$.

6. Modus tercapai pada $\mu = \frac{0,3989}{\sigma}$
7. Kurva mendekati nol secara *asimptotis* semakin x jauh dari $x=\mu$.
8. Luas daerah grafik sama dengan satu satuan persegi.

Pada penerapan untuk deteksi kulit manusia, *normal probability density function* digunakan dengan cara menghitung probabilitas setiap *pixel* terhadap sampel yang telah diketahui sebagai kulit manusia. Semakin tinggi nilai probabilitas suatu *pixel* maka berarti *pixel* tersebut memiliki kemiripan yang lebih tinggi terhadap sampel. Dengan demikian maka *pixel* yang memiliki nilai probabilitas lebih tinggi dari *threshold* yang ditentukan dapat diduga/diklasifikasikan sebagai kulit manusia.