



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

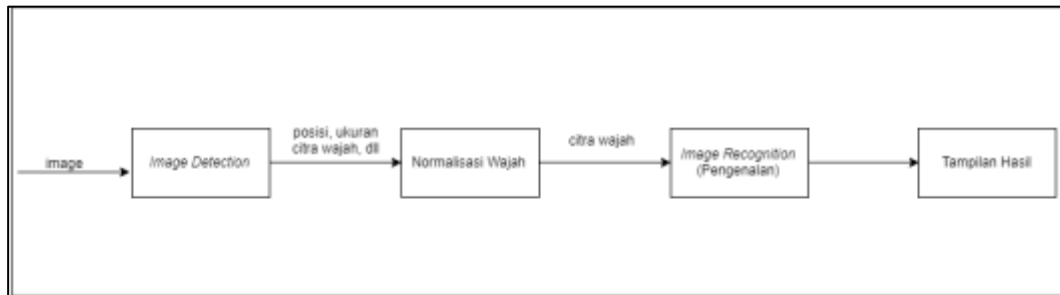
2.1 Face Recognition

Face Recognition adalah cara mengenali wajah manusia melalui teknologi. Sistem *Face Recognition* menggunakan *biometric* untuk memetakan fitur wajah baik dari sebuah foto maupun video dengan membandingkan informasi pada *database* wajah untuk menemukan sebuah kecocokan. *Face Recognition* merupakan sebuah motivasi penelitian pengembangan dari pengenalan wajah termasuk dalam lingkup *biometric*, pengawasan, interaksi manusia dan komputer, dan *multimedia management* (Li dan Jain, 2005:1).

Manusia pada umumnya mungkin pandai mengenali wajah dan memungkinkan untuk menemukan dan mengidentifikasi wajah seseorang seperti anggota keluarga, teman, atau kenalan dengan mudah karena manusia sudah terbiasa dengan fitur-fitur wajah mereka mulai dari mata, hidung, mulut, dan bagaimana mereka bertemu dan begitulah cara kerja sistem pengenalan wajah. Dimana teknologi dapat melihat dan mengenal wajah dengan melihat data.

Pada umumnya, *face recognition* dilakukan dari sisi depan dengan pencahayaan yang merata ke seluruh wajah. Akan tetapi muncul beberapa permasalahan, seperti posisi wajah, skala, atau jarak wajah, orientasi, umur, ekspresi wajah, dan lain sebagainya. *Face Recognition* pada umumnya mencakup empat modul utama (Li dan Jain, 2005:2), yaitu deteksi, *alignment*, ekstraksi fitur, dan pencocokan. Dimana deteksi wajah adalah langkah awal untuk melakukan

identifikasi wajah atau *face recognition* dan memperoleh akurasi proses *alignment* yang sudah dilakukan normalisasi dengan pengambilan data yang efektif untuk memisahkan citra wajah dengan orang-orang yang berbeda satu sama lain. Gambar 2.1 menunjukkan proses pengenalan wajah secara umum yang dilakukan pada *face recognition*.



Gambar 2.1 Proses Pengenalan Wajah Secara Umum (Li dan Jain,2005)

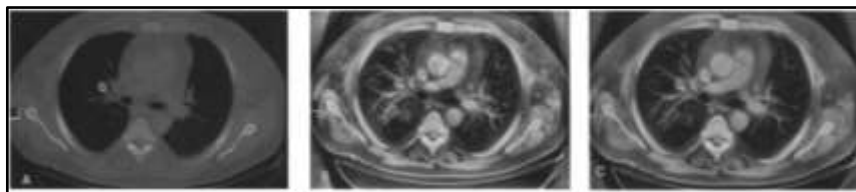
Pada *image detection* dilakukan biasanya dilakukan pendeteksian wajah, dimana sistem akan mencari karakteristik wajah dengan menggunakan sistem *matching* misalnya melakukan *edge detection* pada bagian tertentu, misalnya pada mata, maka hasil berupa garis terluar mata dari sebuah gambar. Banyak hal yang menjadi masalah dalam *face detection*, misalnya seperti posisi wajah yang cukup bervariasi serta ekspresi wajah yang beragam. Setelah itu sistem akan melakukan normalisasi sebelum melakukan pengenalan autentikasi wajah, semua wajah gambar tersebut biasanya dengan tujuan untuk mengurangi efek yang disebabkan keragaman iluminasi biasanya gambar wajah tersebut akan di- *rescale*, diputar, atau dipindahkan dengan berbagai teknik dan metode yang ada dalam *face recognition*. Normalisasi wajah dilakukan karena salah satu permasalahan yang sering dijumpai pada sistem pengenalan wajah adalah adanya variasi iluminasi atau pencahayaan serta ukuran gambar pada citra wajah yang disebabkan karena perbedaan posisi dan sumber cahaya saat proses pengambilan citra wajah sehingga pengaruh iluminasi

cukup berpengaruh terhadap variabilitas citra wajah orang yang sama, yang berbeda pencahayaannya selalu lebih besar dibandingkan variabilitas citra orang yang berbeda (Adini dkk., 1997:721-732).

2.2 Contrast Limited Adaptive Histogram Equalization

Contrast Limited Adaptive Histogram Equalization (CLAHE) adalah sebuah generalisasi dari metode *Adaptive Histogram Equalization* (AHE). Cara kerja algoritma CLAHE berbeda dengan *Histogram Equalization* (HE) yang bekerja secara global. CLAHE membagi citra ke dalam area-area yang lebih kecil dan menerapkan HE untuk masing-masing area tersebut dan meningkatkan *contrast adaptive* yang didasarkan oleh *Adaptive Histogram Equalization*, dimana histogram dihitung berdasarkan wilayah *pixel* (Zuiderveld, dkk, 1994).

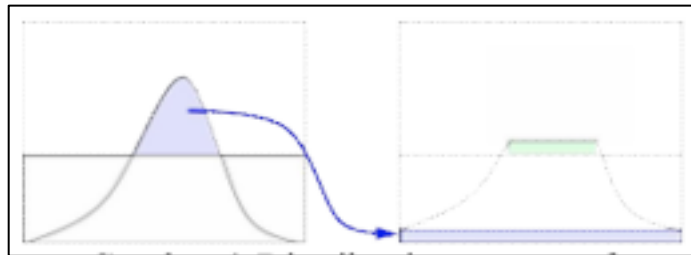
Dalam berbagai penelitian algoritma CLAHE memiliki kemampuan untuk mendeteksi perbedaan *contrast* yang selektif (Zimmerman, 1989) menunjukkan hasil studi ROC pada CT *scan* bahwa CLAHE memperlihatkan *contrast* yang lebih *detail* dibandingkan dengan AHE. Gambar 2.2 menunjukkan perbandingan gambar CT yang dihasilkan oleh algoritma AHE dan CLAHE (Zimmerman, 1989).



Gambar 2.2 Hasil Gambar Perbandingan Algoritma AHE dan CLAHE. (a) Citra Asli. (b) *Image* dengan AHE. (c) *Image* dengan CLAHE. (Zimmerman, 1989)

Algoritma CLAHE menggunakan dua buah parameter: *block size* (N) dan *clip Limit* (CL) untuk mengontrol kualitas sebuah *image*. Biasanya dalam melakukan pembatasan *contrast*, algoritma CLAHE digunakan untuk *image* yang memiliki *low-contrast* (Pisano, E., dkk., 1998). *Clip Limit* pada algoritma CLAHE

digunakan untuk mengatasi masalah *noise* dengan memotong histogram pada nilai yang ditentukan sebelum menghitung nilai *Cummulative Distribution Function* (CDF). Pada umumnya, nilai histogram yang berada pada nilai *clip limit* dianggap kelebihan (*excess*) *pixel* yang akan didistribusikan kepada area sekitar di bawah *clip limit*, sehingga histogram merata. Gambar 2.3 menunjukkan gambar Distribusi *excess pixel* pada *histogram* (Kanditami, dkk,2014).



Gambar 2.3 Distribusi Excess pixel pada histogram (Kanditami, dkk, 2014)

Clip histogram dapat diterapkan secara otomatis menyesuaikan level *clip* dan bergantung dengan *enhancement background* gambar. Algoritma CLAHE biasanya menggunakan *Rayleigh Distribution* untuk melakukan *clip histogram* yang terdapat pada Rumus 2.1.

$$Rayleigh\ g = g_{min} + [2 \alpha^2 \ln \left(\frac{1}{1 - P(f)} \right)]^{0.5} \quad (2.1)$$

Variabel g_{min} adalah nilai minimum dari *pixel*, $P(f)$ adalah nilai *cumulative probability distribution*. Pada umumnya metode CLAHE menerapkan fungsi histogram berdasarkan setiap wilayah dengan memotong *pixel* dan mendistribusikan ke nilai *gray-scale* sehingga setiap gambar asli memiliki nilai *gray-scale minimum* dan *gray-scale maximum* yang sama. Metode CLAHE yang digunakan untuk meningkatkan gambar asli dengan membagi gambar intensitas asli menjadi beberapa wilayah bagian. Total jumlah wilayah seukuran dengan $M \times N$ dan biasanya menggunakan ukuran 8×8 untuk tetap menjaga kualitas *image asli*.

Lalu algoritma CLAHE akan menghitung *histogram* setiap *region* sesuai dengan nilai *gray-scale* pada *array image*. Kemudian menghitung nilai *contrast limited histogram region* sesuai dengan nilai *clip limit* yang ditetapkan pada Rumus 2.2

$$N_{avg} = \frac{Nr_x \times Nr_y}{N_{gray}} \quad (2.2)$$

N_{avg} adalah jumlah *pixel*, N_{gray} adalah angka *level gray-scale* yang ada pada *region* yang telah ditentukan, Nr_x dan Nr_y merupakan *pixel* yang ada pada matriks X dan Y dari *region*. Sedangkan rumus *Clip Limit* ditetapkan sebagai berikut

$$N_{CL} = N_{clip} \times N_{avg} \quad (2.3)$$

N_{CL} adalah nilai dari *Clip Limit* dan N_{clip} adalah nilai normalisasi *Clip Limit* dengan *range* [0, 1]. Jika angka pada gambar lebih besar dari N_{CL} maka N_{clip} bernilai satu, dan total dari *clipped pixel* didefinisikan sebagai $N_{\Sigma clip}$, lalu rata-rata *gray level* dirumuskan dengan

$$N_{avggray} = \frac{N_{\Sigma clip}}{N_{gray}} \quad (2.4)$$

Histogram Clipping Rule memiliki aturan sebagai berikut:

$$\text{If } H_{region}(i) > N_{CL} \text{ then } H_{region_{clip}}(i) = N_{CL} \quad (2.5)$$

$$\text{Else if } (H_{region}(i) + N_{avggray}) > N_{CL} \text{ then } H_{region_{clip}}(i) = N_{CL} \quad (2.6)$$

$$\text{Else } H_{region_{clip}}(i) = H_{region}(i) + N_{CL} \quad (2.7)$$

Dimana $H_{region_{clip}}(i)$ dan $H_{region}(i)$ adalah *clip histogram* dan *original histogram* dari setiap *i-gray level*. Lalu Algoritma CLAHE akan melakukan

distribusi ulang setiap *pixel* yang tersisa sampai semua *pixel* didistribusikan dengan rumus yang terdapat pada Rumur 2.8

$$step = \frac{N_{gray}}{N_{remain}} \quad (2.8)$$

N_{remain} merupakan sisa *pixel* yang belum diselesaikan dan hasil dari nilai *step* merupakan bilangan *positive* setidaknya satu. Pada langkah ini, algoritma CLAHE memulai mencari *minimum level* dan *maximum level* dari *gray scale*. Jika jumlah *pixel* $< N_{CL}$ maka nilai *pixel* akan didistribusikan ke satu *gray scale level*. Algoritma CLAHE akan melakukan *looping* sampai semua *pixel* telah terdistribusi. Algoritma CLAHE akan meningkatkan intensitas di setiap *region* dengan *Rayleigh Transformation* yang menjadikan probabilitas kumulatif sehingga gambar terlihat lebih natural yang diterapkan seperti Rumus 2.1. (Ma, dkk., 2017).

Besaran nilai *region size* dan *clip limit* pada algoritma CLAHE cukup berpengaruh pada kualitas *contrast image*. Maka dari itu, dalam penelitian ini, digunakan *region size* dan *clip limit* dari penelitian sebelumnya dari jurnal yang berjudul “Analisis *Contrast Limited Adaptive Histogram Equalization* (CLAHE) dan *Region Growing* Dalam Deteksi Gejala Kanker Payudara pada Citra Mammogram” (Kanditami, dkk.,2014) dengan menggunakan nilai *Region Size* berukuran 2, 4, 8, dan 16. Tabel 2.1 yang menunjukkan tabel penelitian pada Algoritma CLAHE.

Tabel 2.1 Tabel Penelitian Region Size dan Clip Limit

<i>Region Size</i>	<i>Clip Limit</i>
2	0.001
	0.002
	0.005
	0.009
	0.02
	0.2
	0.5
	0.9
4	0.001
	0.002
	0.005
	0.009
	0.02
	0.2
	0.5
	0.9
8	0.001
	0.002
	0.005
	0.009
	0.02
	0.2
	0.5
	0.9
16	0.001
	0.002
	0.005
	0.009
	0.02
	0.2
	0.5
	0.9

2.3 K-Nearest Neighbors

K-Nearest Neighbors merupakan suatu algoritma yang berfungsi untuk melakukan klasifikasi yang termasuk dalam *supervised learning* dimana algoritma mengklasifikasi suatu data berdasarkan pembelajaran (*train datasets*) yang kemudian mengambil nilai K tetangga terdekatnya (*nearest neighbors*) dengan K merupakan banyaknya tetangga terdekat dengan menggunakan *Memory-based*

Classification sebagai *training example* saat *run time* (Alpaydin.E., 1997) dengan menggunakan *Euclidean Distance* untuk mengambil *training samples* dan menghitung jarak antar tetangga.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.9)$$

Euclidean Distance merupakan salah satu metode perhitungan jarak yang digunakan untuk mengukur jarak dari 2 buah titik dalam *Euclidean Space*. *Euclidean distance* digunakan untuk mencerminkan ketidaksamaan pola pixel dengan pengukuran skala *image* yang sama (Jannah, 2010). Setiap *pixel* pada wajah pasti mewakili informasi yang unik sehingga penelitian ini berdasarkan pada *pixel classification* dengan mengubah suatu *pixel matrix* menjadi *vector* sebelum melakukan *classification* dengan *Euclidean Distance*. Nilai K yang digunakan dalam penelitian ini berdasarkan jurnal sebelumnya yang berjudul *Implementation of K-Nearest Neighbors Face Recognition on Low-power Processor* (Setiawan, Eko dan Adharul Muttaqin, 2013) yang menyimpulkan bahwa nilai $k = 1$ pada algoritma K-Nearest Neighbors pada *face recognition Low-power processor* menghasilkan akurasi terbaik yaitu 91,5%. Maka untuk melakukan variasi penelitian pada nilai k digunakan $K = (1,3, 5,7,18,25, \text{ dan } 30)$ untuk membandingkan hasil akurasi terbaik dari nilai K-Nearest Neighbors. Setelah itu untuk perhitungan persentase akurasi hasil prediksi digunakan rumus

$$\text{Hasil Akurasi} = \frac{\text{Jumlah data prediksi benar}}{\text{Jumlah data testing}} \times 100\% \quad (2.10)$$