

COMPUTING GRAYSCALE OF FACE DETECTION MENGUNAKAN METODE SOBEL DAN LAPLACIAN OF GAUSSIAN

Kustanto

Jurusan Informatika Fakultas Teknologi Informasi, Universitas Islam Balitar Blitar

email : kustanto@unisbablitar.ac.id

Abstrak : Citra dari sudut pandang matematis, merupakan fungsi menerus atau continue dari intensitas cahaya pada bidang dua dimensi $f(x, y)$, dengan x dan y adalah koordinat spasial dan amplitude f pada pasangan koordinat (x, y) yang disebut intensitas atau derajat keabuan citra pada titik tersebut. Jika x, y , dan semuanya berhingga, dan nilainya diskrit, citra tersebut merupakan citra digital. Citra digital adalah citra yang disimpan dalam format digital atau dalam bentuk file. "Pengolahan citra digital merupakan proses yang bertujuan untuk memanipulasi dan menganalisis citra digital dengan bantuan komputer". Penggunaannya, deteksi wajah menggunakan operator berbasis turunan pertama dan turunan kedua. Saat ini telah ada beberapa metode yang dapat digunakan dalam mendeteksi wajah, contohnya adalah metode Sobel dan Laplacian of Gaussian atau LoG, kedua metode ini cocok dalam menyelesaikan penghitungan komputasi. Kedua metode deteksi wajah ini menghasilkan citra yang berbeda, dimana citra yang dihasilkan metode Sobel lebih baik dan jumlah pixel warna putih yang diperoleh lebih banyak dibandingkan dengan metode Laplacian.

Kata kunci : Citra, Deteksi wajah, Laplacian of Gaussian, komputasi, dan Sobel

I. PENDAHULUAN

Citra atau image, istilah lain untuk gambar, sebagai salah satu komponen multimedia yang berperan sangat penting sebagai bentuk informasi visual. Citra dari sudut pandang matematis, merupakan fungsi menerus atau continue dari intensitas cahaya pada bidang dua dimensi $f(x, y)$, dengan x dan y adalah koordinat spasial dan amplitude f pada pasangan koordinat (x, y) yang disebut intensitas atau derajat keabuan citra pada titik tersebut. Jika x, y , dan semuanya berhingga, dan nilainya diskrit, citra tersebut merupakan citra digital (Gonzalez et al ; 2004:2). Citra digital adalah citra yang disimpan dalam format digital atau dalam bentuk file. "Pengolahan citra digital merupakan proses yang bertujuan untuk memanipulasi dan menganalisis citra digital dengan bantuan komputer" (Fanzuanta, 2011:1).

Penggunaannya, deteksi wajah menggunakan operator berbasis turunan pertama dan turunan kedua. Saat ini telah ada beberapa metode yang dapat digunakan dalam mendeteksi wajah, contohnya adalah metode Sobel dan Laplacian of Gaussian atau LoG, dan sebagainya. Pembahasan dua buah metode deteksi wajah, yaitu metode Sobel dan Laplacian, karena kedua metode ini lebih mudah dalam menyelesaikan penghitungannya. Kedua metode deteksi wajah ini menghasilkan citra yang berbeda, dimana citra yang dihasilkan metode Sobel lebih

baik dan jumlah pixel warna putih yang diperoleh lebih banyak dibandingkan dengan metode Laplacian.

Metode Laplacian merupakan metode yang menggunakan operator Laplacian Operator Laplacian adalah operator yang berbasis gradien yang menggunakan dua buah kernel yang berukuran 3x3 pixel. Operator ini mengambil arah diagonal untuk penentuan arah dalam penghitungan nilai gradient. Sedangkan metode Sobel merupakan metode yang menggunakan operator Sobel. Operator ini menggunakan dua buah kernel yang berukuran 3x3 pixel untuk penghitungan gradiennya.

II. METODE PENELITIAN

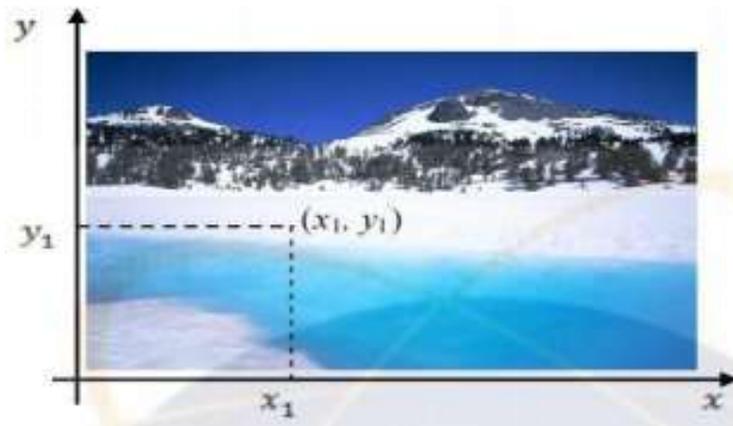
Deteksi wajah dapat dipandang sebagai masalah klasifikasi pola dimana inputnya adalah citra masukan dan akan ditentukan output yang berupa label kelas dari citra tersebut. Dalam hal ini terdapat dua label kelas, yaitu wajah dan non-wajah. Teknik-teknik pengenalan wajah yang dilakukan selama ini banyak yang menggunakan asumsi bahwa data wajah yang tersedia memiliki ukuran yang sama dan latar belakang yang seragam. Di dunia nyata, asumsi ini tidak selalu berlaku karena wajah dapat muncul dengan berbagai ukuran dan posisi di dalam citra dan dengan latar belakang yang bervariasi. Pendeteksian wajah (*face detection*) adalah salah satu tahap awal yang sangat penting sebelum dilakukan proses pengenalan wajah (*face detection*).

Penelitian dari Nugroho dan kawan-kawan, mengelompokkan metode deteksi wajah menjadi empat kategori, yaitu :

1. *Knowledge-based method*. Metode ini kebanyakan digunakan untuk lokalisasi wajah,
2. *Feature invarian approach*. Metode ini kebanyakan digunakan untuk lokalisasi wajah,
3. *Template matching method*. Metode ini digunakan untuk lokalisasi wajah maupun deteksi wajah,
4. *Appearance-based method*. Metode ini kebanyakan digunakan untuk deteksi wajah.

Citra Digital

Citra dapat dikatakan sebagai citra digital jika citra tersebut disimpan dalam format digital (dalam bentuk *file*). Seperti halnya proses digitisasi dalam bentuk data lain, proses digitisasi pada data citra juga merupakan proses pengubahan suatu bentuk data citra dari yang bersifat analog ke dalam bentuk data digital, yang mana proses ini dapat dilakukan dengan alat bantu, yang salah satunya berupa kamera. Hanya citra digital yang dapat diolah menggunakan komputer. Jenis citra lain jika akan diolah dengan komputer harus diubah dulu menjadi citra digital.



Gambar 2.1 Citra Digital(Citra Annisa, 2010)

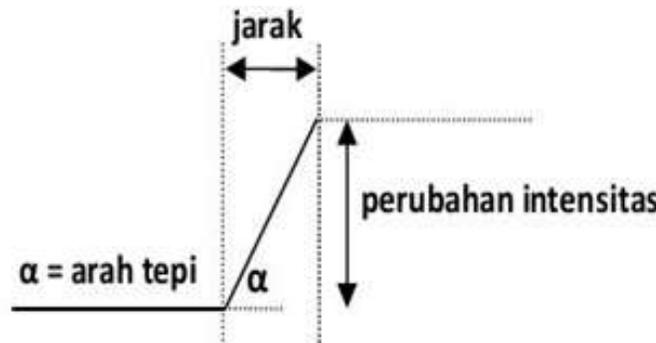
Gambar 2.1 ditampilkan dengan matriks berukuran $N \times M$ (N menyatakan baris atau tinggi, M menyatakan kolom atau lebar) (Hestningsih, 2008).

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (1)$$

Keterangan:

N = jumlah baris, $0 \leq y \leq N - 1$ M = jumlah kolom, $0 \leq x \leq M - 1$
 L = maksimal warna intensitas (derajat keabuan), $0 \leq f(x,y) \leq L - 1$

Citra digital biasanya berbentuk persegi panjang, secara visualisasi dimensi ukurannya dinyatakan sebagai lebar \times tinggi. Ukurannya dinyatakan dalam titik atau pixel (*pixel = picture element*) dan dapat pula dinyatakan dalam satuan panjang (mm atau inci = *inch*).



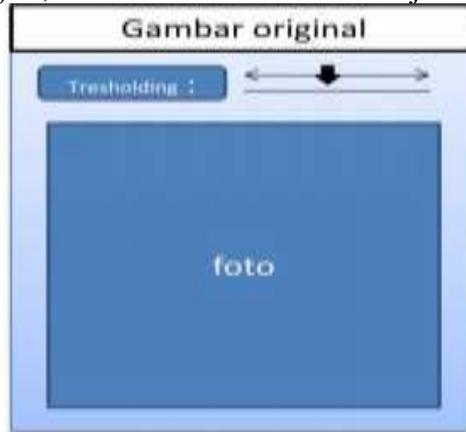
Gambar 2.11 Model Tepi Satu Dimensi (Febriani, 2008)

Deteksi Tepi (*Edge Detection*)

Secara umum tepi dapat didefinisikan sebagai batas antara dua *region* atau dua piksel yang saling berdekatan yang memiliki perbedaan intensitas yang tajam atau tinggi. Tepi dapat diorientasikan dengan suatu arah, dan arah ini berbeda-beda, tergantung Pada perubahan intensitas. Untuk lebih memahami defenisi tepi, gambar dibawah ini memperlihatkan model tepi dalam ruang satu dimensi (Febriani, 2008).

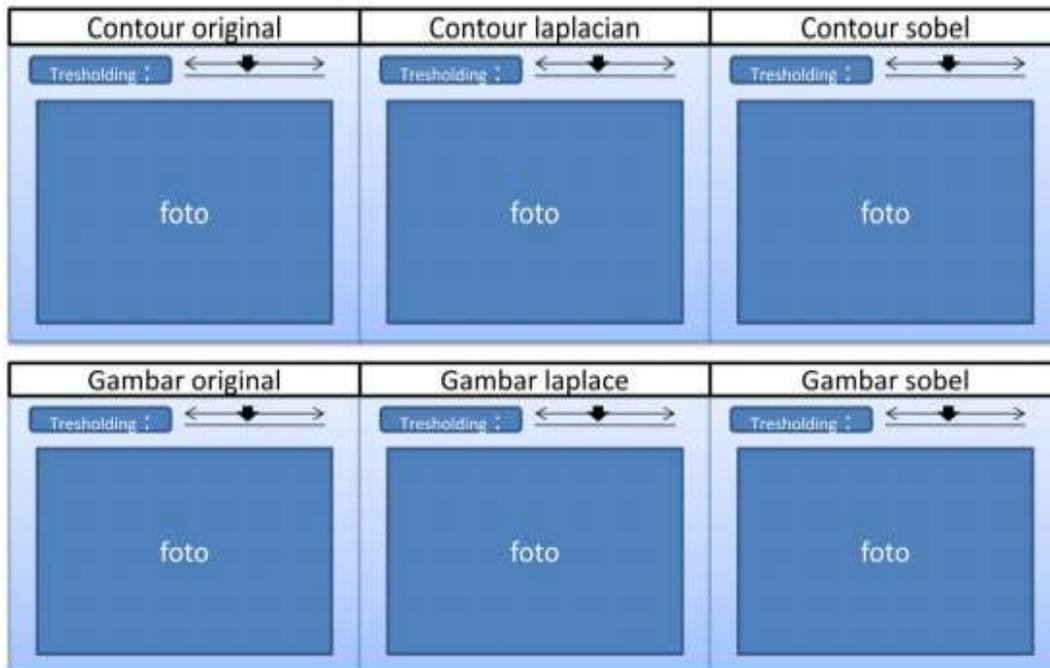
**Langkah-Langkah Proses Pengimplementasian Aplikasi :
Rancangan Interface :**

1. Halaman utama digunakan untuk mengambil foto dari *file*, Form hasil: gambar atau foto yang akan diuji cobakan terhadap programnya Setelah berhasil mengambil foto dari *file*, maka foto tersebut akan diuji cobakan pada programnya.



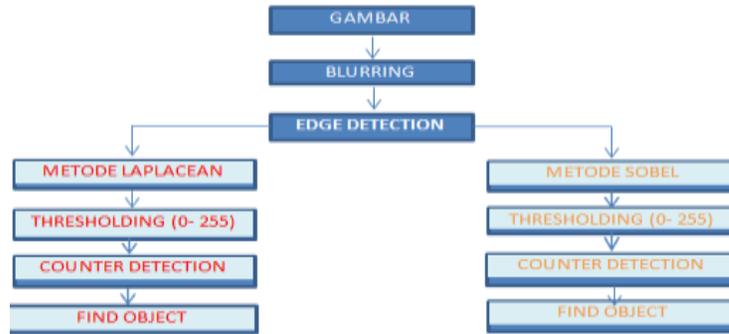
Gambar 3.1 halaman utama

2. Form hasil: hasil gambar atau foto yang telah diuji cobakan. Setelah foto diuji, maka akan terlihat perbandingan antara kedua metode *Sobel* dan *Laplacian of Gaussian (LoG)*. Jadi antar kedua metode tersebut akan terlihat *noise* dan garis batas geometrinnya *dicontours*.



Gambar 3.3 form hasil yang diuji antara metode *Sobel* dan *Laplacian of Gaussian*

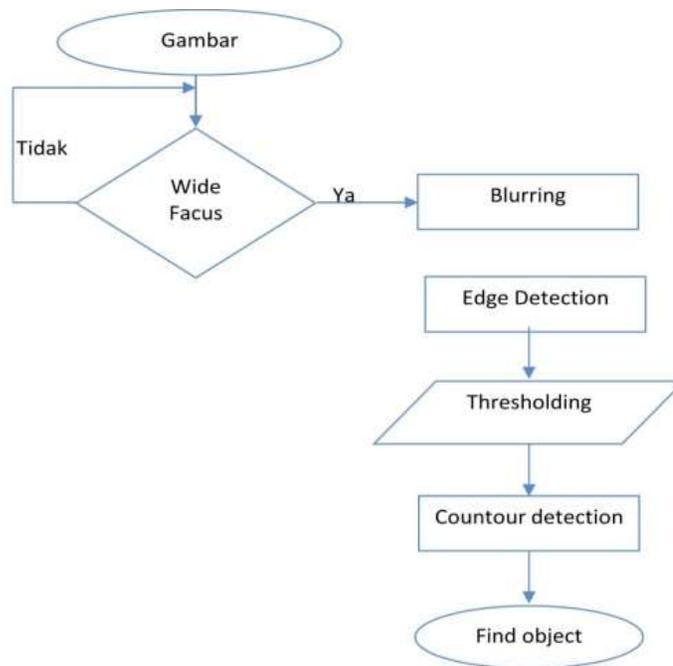
Berikut penjelasan langkah-langkah proses pengimplementasian pembuatan aplikasi ini . Untuk proses secara umum bisa digambarkan dengan diagram blok seperti berikut :



Gambar 3.10 Diagram Blok Proses Secara Umum

Pendefinisian, gambar akan menuju ke-*blurring* maka sebuah gambar antara metode *laplacean of Gaussian* dan *sobel* dengan thresholding antara 0-255 dan pendeteksian *contours* antara 2 metode tersebut kemudian munculah perbedaan gambar dengan menggunakan metode *laplacean of gaussian* dan metode *sobel* tersebut. Jadi unsure dari prosesnya dari kedua metode tersebut metode *Sobel* lebih akurat dibandingkan dengan metode *Laplacian of Gaussian*.

Diagram Blok



Gambar 3.10 Diagram Blok Proses

Jika gambar ke *wide focus* benar maka lanjut ke *blurring*, jika gambar ke *wide focus* salah maka akan kembali lagi ke gambar, jika gambar benar maka akan lanjut ke *blurring* maka ke *edge detection*, lanjut lagi ke *thresholding* ke *countour detection* setelah semua benar maka akan ke *find object*. Average filter pada sebuah *filter lowpass* yang menghitung rata-rata, abu-abu (*grayscale*). Filter ini menggantikan nilai setiap *pixel* [i, j] dengan rata-rata dari *pixel* di beberapa *pixel* tetangga di sekitarnya, yaitu jumlah bobot dari [i + p, j + q], dengan p = -k ke k, q = -k ke k untuk beberapa nilai k positif, bobot non-negatif dengan bobot tertinggi pada p = q = 0. Penggunaan metode *average filter* diharapkan mendapatkan nilai yang rata dan sama pada tiap *pixel* tetangganya yang dihasilkan. Jika menggunakan metode *filter* lain, misalkan median *filter* atau *Gaussian filter*, mendapatkan proses yang berdasarkan nilai tengahnya pada tiap kernel tetangganya. Sehingga nilai tengah pada *median filter* atau *Gaussian filter* lebih besar dibandingkan dengan nilai tetangganya. Selain itu, median *filter* atau *Gaussian filter* kebanyakan berfungsi sebagai *noise reduction* atau mengurangi *noise* pada gambar.

Edge Detection Yaitu proses deteksi tepi pada suatu gambar yang menghasilkan gambar berupa garis garis tepi pada *objek* yang ada pada gambar. Metode untuk edge detection beragam. Pada aplikasi ini dengan menggunakan metode Sobel yaitu dengan memanfaatkan kernel 3x3 sebagai berikut:

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2)$$

Pada matriks (2) ditampilkan sebagai filter vertikal.

$$\begin{bmatrix} -1 & 0 & 1 \\ 2 & & \end{bmatrix} \quad (3)$$

Pada matriks (3) ditampilkan sebagai filter horizontal.

Untuk proses secara umum, contoh *source code*-nya seperti berikut :

```

/** @function main */
int main( int argc, char** argv )
{
//initialitation value int scale = 1;
int delta = 0;
int ddepth = CV_16S;
/// Load source image and convert it to gray
src = imread( "C:/gambar/10.jpg", CV_LOAD_IMAGE_COLOR);
/// Convert image to gray and blur it cvtColor( src,
src_gray, CV_BGR2GRAY ); blur( src_gray, src_gray,
Size(3,3) );
//sobel
Mat sobel_x, sobel_y;
Mat abs_sobel_x, abs_sobel_y;
// sobel X
Sobel( src_gray, sobel_x, ddepth, 1, 0, 3, scale, delta,
BORDER_DEFAULT );
convertScaleAbs( sobel_x, abs_sobel_x ); // sobel Y
Sobel( src_gray, sobel_y, ddepth, 0, 1, 3, scale, delta,
BORDER_DEFAULT );
convertScaleAbs( sobel_y, abs_sobel_y );

```

```

/// Total sobel (approximate)
addWeighted( abs_sobel_x, 0.5, abs_sobel_y, 0.5, 0, sobel
);

```

Penggabungan *edge* detection dari 2 gambar yang diproses menjadi 1 gambar utuh untuk diproses lagi pada tahapan selanjutnya. Tampilan *source code* nya seperti berikut :

```

//sobel
Mat sobel_x, sobel_y;
Mat abs_sobel_x, abs_sobel_y;
// sobel X
Sobel( src_gray, sobel_x, ddepth, 1, 0, 3, scale, delta, BORDER_DEFAULT);
convertScaleAbs( sobel_x, abs_sobel_x ); // sobel Y
Sobel( src_gray, sobel_y, ddepth, 0, 1, 3, scale, delta, BORDER_DEFAULT);
convertScaleAbs( sobel_y, abs_sobel_y );
/// Total sobel (approximate)
addWeighted( abs_sobel_x, 0.5, abs_sobel_y, 0.5, 0, sobel );

```

Thresholding merupakan proses konversi citra *grayscale* menjadi citra biner (*binary image*) dengan nilai 1 dan 0. Citra *grayscale* memiliki nilai intensitas warna antara 0-255 sehingga pada untuk memberikan nilai *binary image* pada citra ditentukan terlebih dahulu batas ambang yang akan membagi 2 daerah pada *citra*. *Thresholding* bertujuan menghasilkan citra biner secara matematis dapat dituliskan sebagai berikut :

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{if } f(x, y) < T \end{cases} \quad (4)$$

Dengan $g(x, y)$ adalah citra biner dari citra *grayscale* $f(x, y)$ dan T menyatakan nilai *threshold*.

```

/// Detect edges using Threshold
threshold( abs_dst, threshold_output_laplace, thresh, 255, THRESH_BINARY);
/// Find contours
findContours( threshold_output_laplace, contours_laplace, hierarchy_laplace,
CV_RETR_TREE, CV_CHAIN_APPROX_SIMPLE, Point(0, 0) );

```

Contours Detection adalah garis batas dari bentuk geometric dalam gambar digital. Karena identifikasi kontur sangat penting untuk menganalisis isi gambar, ekstraksi kontur menjadi salah satu bagian penting dalam masalah computer vision dan pengenalan pola atau untuk objek gambar. Algoritma ekstraksi kontur terdapat dua tahapan utama. Tahap pertama adalah preprocessing untuk mengekstrak satu set poin berorientasi dari input gambar. Tahap kedua menentukan kontur diantara titik yang berorientasi menggunakan geometri. Tahap kedua tersebut masih terdiri lagi dari beberapa langkah yaitu titik saring dengan teknik pengelompokkan, selanjutnya titik dihubungkan berdasarkan kedekatan dan orientasi ke jalur yang mewakili kontur, dan akhirnya jalur yang disederhanakan dengan mengurangi jumlah titik yang mereka miliki (Pedro et al, 2009, p 1)

```

/// Find contours
findContours( threshold_output_sobel, contours_sobel, hierarchy_sobel, CV_RETR_TREE,
CV_CHAIN_APPROX_SIMPLE, Point(0,0) );

```

III. HASIL DAN PEMBAHASAN

Citra yang dapat digunakan dalam percobaan untuk kedua metode wajah adalah citra dengan format JPEG (ekstensi *.jpg). Pengujian gambar yang dilakukan dengan membagi gambar dengan kamera – kamera yang berbeda dan metode yang berbeda juga. Seperti menggunakan kamera hp berbagai merek dengan perbedaan pada *pixel* pada kamera tersebut, dan menggunakan kamera digital dan DLSR berbagai merek dan spesifikasi.

Perbandingan Citra Hasil Deteksi Wajah Secara Visual

Semua data uji, citra yang diperoleh dari hasil metode *Sobel* lebih jelas dan garis tepi yang diperoleh lebih banyak terhubung disbanding dengan metode *laplacian* yang masih putus-putus. Dengan menggunakan metode *Sobel*, masih banyak tepi yang bisa terbaca disbanding dengan metode *Laplacian*.

Metode *Sobel* lebih baik dibandingkan metode *Laplacian* karena operator dan ukuran *kernel* yang digunakan berbeda. Metode *Sobel* menggunakan operator *Sobel*:

$$S_x = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \text{ dan } S_y = \begin{matrix} 1 & 2 & 1 \\ -2 & 0 & 2 \\ -1 & -2 & -1 \end{matrix} \text{ dengan ukuran } 3 \times 3 \text{ pixel} . \text{ Sedangkan}$$

Metode *Laplacian of Gaussian* menggunakan operator *Laplacian Of Gaussian*

$$L_x = \begin{matrix} 0 & -1 & 0 \\ 0 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \text{ dan } L_y = \begin{matrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{matrix} \text{ dengan ukuran } kernel \ 3 \times 3 \text{ pixel}$$

Metode *laplacian* lebih menekankan penghitungan gradient arah diagonal sedangkan metode *Sobel* lebih ke arah vertical dan horisontalnya. Ukuran *kernel* yang dipakai dalam deteksi wajah cukup mempengaruhi hasil yang didapat, semakin besar ukuran *kernel* semakin banyak wajah yang diperoleh, terbukti dalam percobaan lain ketika mendeteksi wajah dengan menggunakan *kernel* berukuran 3×3 *pixel* pada citra.

IV. KESIMPULAN DAN SARAN

Berdasarkan pembahasan yang telah diuraikan, maka diambil beberapa kesimpulan:

1. Kualitas citra *grayscale* untuk *face detection*
 - a. hasil Operator *LoG* menghasilkan tepi citra wajah yang sama jelasnya dengan operator *Sobel* dan Operator *Sobel* memproses tepi citra wajah lebih cepat dibandingkan operator *LoG*.
 - b. bergantung pada jumlah cluster yang diberikan dalam proses pengolahan citra, pemilihan kategori citra acuan yang ditentukan dan pengaruh dari masing-masing citra dalam kategori yang sama.
2. Pengujian *face detection* berdasarkan analisa gambar dan analisa counturs yang telah dilakukan dapat disimpulkan bahwa penerapan metode *sobel* lebih baik dibandingkan dengan penerapan metode *Laplacian of Gaussian* dalam pewarnaan citra *face detection*.

V. DAFTAR PUSTAKA

- Viola, P., Jones, M. J. 2004. ROBUST REAL-TIME FACE DETECTION, International Journal of Computer Vision, Kluwer Academic, Netherlands.
- Charan, BP., Sinha ,G.R., 2010, An Adaptive K-means Clustering Algorithm for Breast Image Segmentation, Int. Journal of Computer Applications, 10(4).
- Gonzalez, R.C., Woods, RE., 2008, Digital Image Processing Third Edition, Pearson Prentice Hall, London.
- Putra, D., 2010, Pengolahan Citra Digital, Andi Yogyakarta. Yogyakarta.
- Russ, JC., 2011, The Image Processing Handbook Sixth Edition, CRC Press. United State.
- Sheikh, H.R., Bovik A.C., 2005, Information Theoretic Approaces to Image Quality Assessment. In: Bovik, A.C. Handbook of Image and Video Processing. Elsevier. Srimi, 2009.,
- Nugroho, S. 2004. SISTEM PENDETEKSI WAJAH MANUSIA PADA CITRA DIGITAL, Tesis Program Studi Ilmu Komputer Jurusan MIPA, Universitas Gadjah Mada, Yogyakarta
- Pengolahan Citra: Konsep Dasar, <http://srini.staff.gunadarma.ac.id/Downloads/files/4881/8+Olah+Citra-Konsep+Dasar.pdf>, diakses 9 Maret 2011
- Suhendra, A., 2008, Akbaramin, Muhammad. 2012. Analisa Perbandingan Kinerja Deteksi Tepi Menggunakan Metode Log, Sobel, Dan Canny Terhadap Format File Jpeg Dan Bmp. (Online). (diakses 25 Januari 2017).
- Catatan Kuliah Pengantar Pengolahan Citra, <http://images.analyst71.multiply.com/attachment/0/Rz6WgoKCiQAAfXBe81/Catatan%20Kuliah%20PC%202007.pdf>, diakses 9 Maret 2016