

IMPLEMENTASI ALGORITMA *COLOR FILTERING* PADA APLIKASI GITAR *VIRTUAL*

Asep Nana H.¹⁾, Dewi Rosmala²⁾, M. Ichwan³⁾, Rio Wahyu Indarko⁴⁾

Jurusan Teknik Informatika, Fakultas Teknologi Industri
Institut Teknologi Nasional
Bandung

¹⁾asepnana@itenas.ac.id, ²⁾d_rosmala@itenas.ac.id, ³⁾ichwan@itenas.ac.id, ⁴⁾rio.wahyu1505@gmail.com

Abstrak

Color Filtering adalah suatu teknik pengolahan citra yang dipakai untuk memanipulasi suatu citra berdasarkan warna spesifik. Cara kerjanya adalah dengan membandingkan komponen warna setiap pixel citra dengan warna spesifik. Algoritma *Color Filtering* dapat diimplementasikan untuk aplikasi gitar *virtual* dengan menggunakan kamera dan warna penanda jari tangan sebagai alat bantu. Proses *Color Filtering* digunakan dalam pendeteksian untuk memfilter warna RGB (Red Green Blue). Sistem yang dibangun menyimpan beberapa titik pola sesuai dengan kord gitar dan menyimpan suara kord gitar tersebut. Pengguna harus menggerakkan jari-jari yang diberi penanda warna sesuai dengan posisi titik kord yang tersimpan, kemudian sistem akan memproses dan memfilter warna penanda jari tangan, apabila sesuai, maka sistem akan mengeluarkan suara kord gitar. Hasil pengujian sistem dapat bekerja mendeteksi warna pada resolusi rendah *width* 640 – *height* 480 sampai resolusi tinggi *width* 1280 – *height* 720 dan intensitas cahaya rendah 10 lumen sampai intensitas cahaya tinggi 230 lumen. Dapat disimpulkan bahwa sistem dapat mendeteksi warna pada resolusi *width* 640 – *height* 480 dengan intensitas cahaya 57 lumen.

Abstract

Color Filtering is an image processing techniques used to manipulate an image based on specific color. The way it works is by comparing the color components of each pixel image with a specific color. *Color Filtering* algorithms can be implemented for the virtual guitar app to use the camera and color markers fingers as tools. *Color Filtering* process is

used in the detection of color filter RGB (Red Green Blue). The system is built to save some point in accordance with guitar chord patterns and storing the guitar chords sound. Users must move the fingers are labeled according to the color of the stored position chord point, then the system will process and filter the marker color fingers, as appropriate. Then the system will sound guitar chords. The test results detecting color system can work at low resolution width 640 - height 480 to a high resolution width 1280 - height 720 and lower light intensity of 10 lumens to 230 lumens of light intensity high. It can be concluded that the system can detect color at a resolution of 640 width - height 480 with 57 lumens of light intensity .

Keywords : Kord Guitar, Color Filtering , Digital Image Processing , Camera

I. PENDAHULUAN

Color Filtering merupakan salah satu algoritma yang digunakan dalam bidang pengolahan citra digital. Algoritma ini merupakan *filter* yang digunakan untuk memfilter warna yang diinginkan pada gambar atau citra berdasarkan nilai komponen pada RGB. Kombinasi RGB inilah yang kemudian dijadikan *filter* sebagai penentu sebuah warna diloloskan atau tidak.

Pada aplikasi gitar pada desktop komputer maupun mobile android, permainan gitar secara virtual tidak menggunakan kamera webcam. Bermain gitar tanpa harus menggunakan objek gitar dan hanya menggerakkan jari-jari tangan yang membentuk kord kord gitar dengan menggunakan *webcam* pada laptop atau pada PC. Aplikasi gitar virtual ini memiliki cara

kerja yang sederhana, dengan menggunakan kamera *webcam* yang menampilkan sebuah gambar video dari kamera tersebut. Didalam video aplikasi terdapat *virtual* gitar berupa gambar *string* (senar) yang fungsinya untuk pencocokan jari jari yang membentuk kord terhadap senar gitar. Tetapi sangat sulit untuk menentukan titik pada jari tangan seseorang sebagai penanda untuk pencocokan titik titik kord pada senar gitar yang harus disesuaikan agar dapat menghasilkan suara. Suara gitar yang dikeluarkan berupa suara rekaman gitar dengan format file .wav.

Maka dari itu dibuatsebuah aplikasi sebagai metode pembelajaran untuk simulasi permainan gitar dengan mengimplementasikan algoritma *Color Filtering* untuk menentukan titik pada penanda jari tangan agar dapat dilakukan proses pencocokan terhadap titik kord gitar yang sesuai.

Rumusan Masalah

Rumusan masalah adalah sebagai berikut:

1. Bagaimanamenerapkan tahapan proses *Color Filtering* dalam memfilter warna pada penanda jari tangan sebagai titik penentu sebuah warna untuk pencocokan terhadap titik kord gitar.
2. Bagaimana mengimplementasikan aplikasi gitar*virtual* menggunakan *webcam*.

Tujuan

Adapun tujuan penelitian yaitu penerapan *Color Filtering* untuk mendeteksi penanda warna pada jari tangan sebagai pencocokan terhadap titik pada kord gitar agar menghasilkan suara kord dalam pembangunan aplikasi gitar *virtual*.

Batasan Masalah

Dalam penelitian ini diberikan batasan antarlain:

1. Kord – kord dasar yang dimainkan adalahA, B, C, E, F, G.
2. Aplikasi gitar ini memperagakan bentuk jenis kord – kord dasar pada gitar yang dilakukan oleh gerakan jari tangan yang diberi penanda warna (solatip) didepan kamera *webcam*.
3. Warna background atau ruangan yang digunakan berbeda dengan penanda warna pada jari.

4. Warna penanda jari tangan sama dengan warna yang dipilih pada saat pemilihan warna yang akan difilter.
5. Tangan yang digunakan untuk pergerakan jari adalah tangan kiri (normal).
6. Pada proses simulasi pergerakan jari, sistem menampilkan senar (*string*) gitar, tidak menampilkan gambar gitar.
7. Senar (*string*) gitar tidak bisa dipetik dan sistem hanya mengeluarkan suara kord.

Kajian Literatur

***Color Filtering*^[1]**

Color Filtering adalah suatu teknikpengolahan citra yang yang dipakai untukmemanipulasi suatu citra berdasarkan warnaspesifik. Cara kerjanya adalah dengan membandingkan komponen warna setiap *pixel*citra dengan warna spesifik.Apabila warnanyasesuai dengan warna spesifik komponen warnapixel tersebut dibiarkan saja.Namun, bilawarnanya tidak sesuai dengan warna spesifikmaka komponen warna *pixel* tersebut diubahmenjadi warna *background*, biasanya menjadiwarna hitam.Warna yang digunakan dalam *Color Filtering* dapat direpresentasikan dalam berbagai ruang warna.

***Grayscale*^[2]**

GrayScale adalah suatu citra dimana nilai dari setiap *pixel* merupakan sampel tunggal. Citra yang ditampilkan adalah citra keabuan dimana intensitasnya berada pada interval 0 – 255, warna hitam (0) pada bagian yang intensitasnya terlemah dan warna putih(255) pada intensitas terkuat. Proses konversi dari citra berwarna menjadi Grayscale menggunakan koefisien dari ITURecommendation BT.709.

***Deteksi Tepi*^[2]**

Deteksi tepi pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra. Tepi-tepi ini akan menandai bagian detail citra. Tepi-tepi pada gambar tersebut terletak pada titik-titik yang memiliki perbedaan tinggi. Terdapat beberapa metode untuk perhitungan deteksi tepi diantaranya metode sobel.

Metode Sobel merupakan pengembangan metode robert dengan menggunakan *filter* HPF yang diberi satu angka nol penyangga. Metode ini mengambil

prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode sobel ini adalah kemampuan untuk mengurangi *noise* sebelum melakukan perhitungan deteksi tepi.

Operator sobel

Peninjauan pengaturan *pixel* di sekitar *pixel*nya (x,y) adalah :

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_3 & (x,y) & a_4 \\ a_5 & a_6 & a_7 \end{bmatrix} \dots \dots \dots (1)$$

Operator sobel adalah magnitudo dari gradien yang dihitung dengan :

$$M = \sqrt{S_x^2 + S_y^2} \dots \dots \dots (2)$$

Dengan konstanta c = 2. Dalam bentuk mask, s_x dan s_y dapat dinyatakan sebagai :

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \dots \dots (3)$$

Arah tepi dihitung dengan persamaan:

$$\alpha(x,y) = \tan^{-1} \left(\frac{S_y}{S_x} \right) \dots \dots \dots (4)$$

Pusat Massa Obyek (Center Of Gravity)^[3]

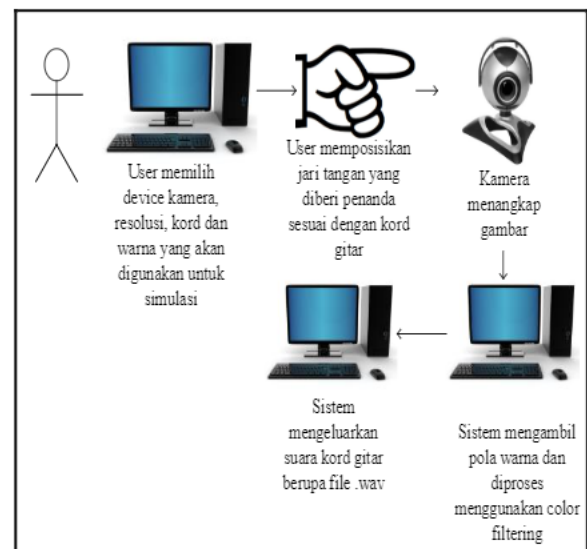
Center of Gravity adalah metode yang dipergunakan untuk menentukan titik keseimbangan dari grafik yang merupakan hasil dari proses pengolahan citra. Persamaan 3 merupakan perumusan matematis dari *center of gravity*.

$$\text{Posisi } X = \left(\frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \right) \dots \dots \dots (5)$$

$$\text{Posisi } Y = \left(\frac{y_1 + y_2 + y_3 + \dots + y_n}{n} \right) \dots \dots \dots (5)$$

Proses Bisnis Gitar Virtual

Gambar 3 menjelaskan proses bisnis yang diawali dengan pergerakan jari tangan yang diberi penanda titik warna. Pergerakan jari tangan ditangkap oleh kamera lalu diproses oleh sistem. Sistem memfilter titik warna pada jari tangan terhadap titik kord yang tampil pada aplikasi dengan menggunakan *Color Filtering*. Apabila gerakan jari tangan dan penanda warna sesuai dengan titik kord pada sistem, maka sistem akan mengeluarkan suara kord gitar sesuai dengan titik kord tersebut. Tetapi, apabila pergerakan jari tangan tidak sesuai dengan titik kord, maka sistem tidak mengeluarkan suara apapun.

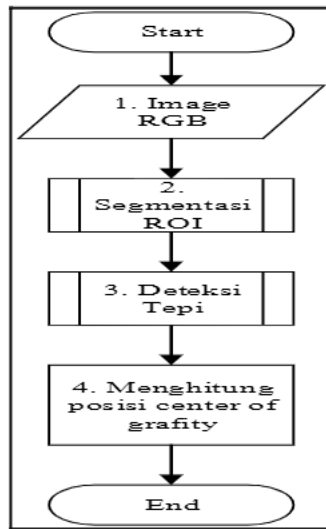


Gambar 3 Proses Bisnis

Proses Color Filtering

Dalam pembuatan aplikasi gitar virtual ini diterapkan sebuah proses *Color Filtering*. Cara kerjanya adalah dengan membandingkan komponen warna setiap *pixel* citra dengan warna spesifik. Apabila warnanya sesuai dengan warna spesifik komponen warna *pixel* tersebut dibiarkan saja. Namun, bila warnanya tidak sesuai dengan warna spesifik maka komponen warna *pixel* tersebut diubah menjadi warna *background*, biasanya menjadi warna hitam. Dalam proses *Color Filtering* terdapat beberapa proses didalam yaitu segmentasi ROI (*Region Of Interest*), deteksi tepi dan menghitung COG (*Centre Of*

Gravity). Berikut merupakan tahapan dari proses *Color Filtering*.



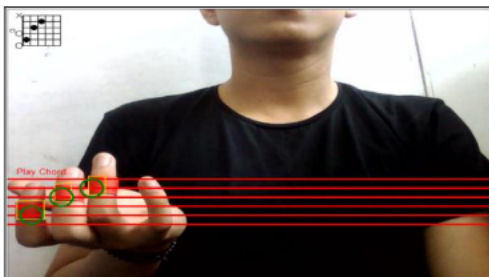
Gambar 4 *Flowchart Color Filtering*

Flowchart pada gambar 4 akan dijelaskan sebagai berikut:

1. Masukkan *image RGB* oleh user yang merupakan pergerakan jari tangan.
2. Proses segmentasi ROI yang menghasilkan gambar hitam putih.
3. Proses deteksi tepi untuk menghasilkan tepi – tepi dari gambar.
4. Menghitung posisi COG untuk menentukan titik tengah dari gambar.

Image RGB

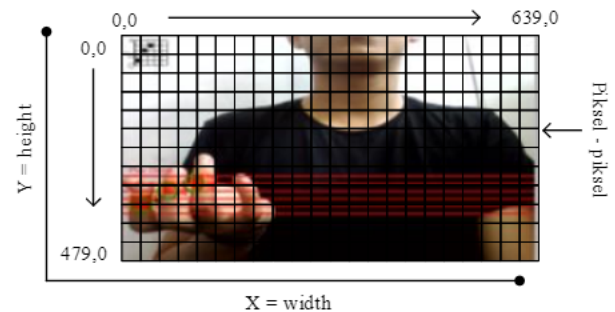
Gambar 5 merupakan masukkan dari warna penanda jari tangan dan user mulai memposisikan jari sesuai dengan titik kord berada. Sistem mendeteksi warna pada penanda jari tangan.



Gambar 5 Memposisikan Jari

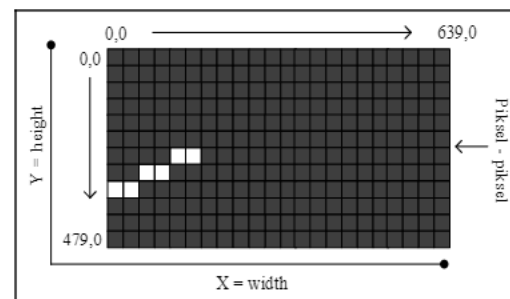
Proses Segmentasi ROI (*Region Of Interest*)

Pada proses segmentasi ROI merupakan proses mengambil *pixel – pixel* yang mendekati warna RGB diubah menjadi putih dan yang lainnya diubah menjadi hitam atau *grayscale*. Pada besaran resolusi terdapat *width* dan *height*, diinisialisasikan *width* = x dan *height* = y. Setiap *pixel* dari *image RGB* dan nilai RGB *marker* akan diubah menjadi hitam putih. Misalkan resolusi *width* = 640, *height* = 480.



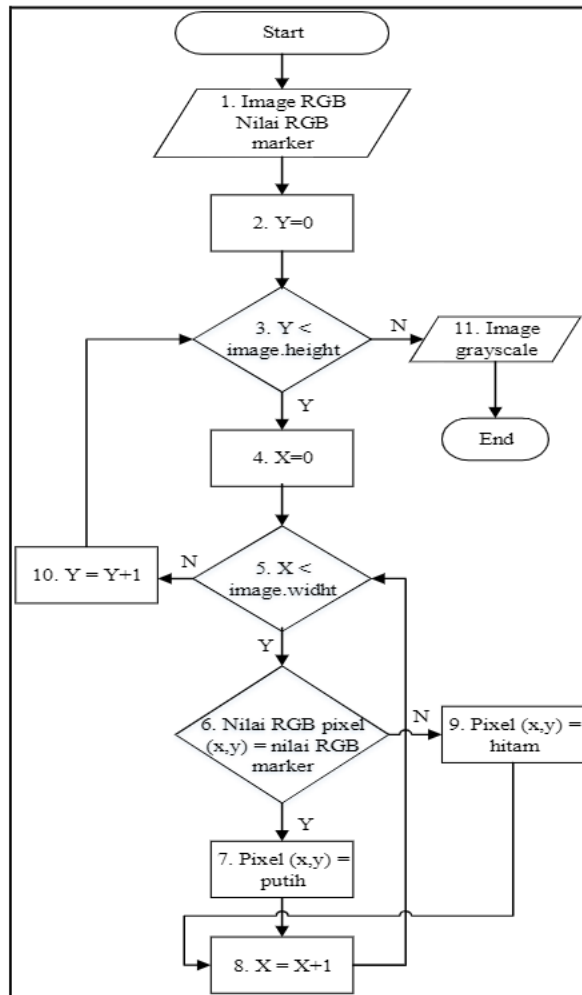
Gambar 6 Proses Segmentasi

makapixel dari *width* 0 – 639 dan *pixel* dari *height* = 479 akan diubah menjadi warna hitam setiap *pixel*nya. *Pixel* yang memiliki nilai RGB *marker* akan diubah menjadi putih.



Gambar 7 Proses Segmentasi

Pada gambar 8 merupakan *flowchart* dari proses segmentasi.



Gambar 8 Proses Segmentasi ROI

Flowchart pada gambar 8 akan dijelaskan sebagai berikut:

1. Masukkan *image* RGB dan nilai RGB *marker* (penanda jari).
2. Proses mulai dari *pixel* Y = 0 (*y pixel* dari *height*).
3. Jika $y < \text{image height}$, jika ya lakukan proses langkah 5, jika tidak lakukan langkah 11.
4. Proses mulai dari *pixel* X = 0 (*x pixel* dari *width*).
5. Jika $x < \text{image width}$, jika ya lakukan langkah 7, jika tidak lakukan langkah 10.

6. apakah nilai RGB *pixel* = nilai RGB *marker*, jika ya lakukan langkah 8, jika tidak lakukan langkah 9.
7. *pixel* (x,y) = putih.
8. $X = X + 1$, kembali ke langkah 6.
9. *Pixel* (x,y) = hitam.
10. $Y = Y + 1$, kembali ke langkah 4.
11. Keluaran *image* grayscale.

Proses Deteksi tepi

Pada gambar 5 merupakan proses dari deteksi tepi. Gambar hasil dari segmentasi ROI telah dihitung setiap *pixel*nya sehingga menghasilkan tepi – tepi dari gambar tersebut.

Pixel – pixel dari hasil grayscale dihitung dengan metode sobel.

Persamaan perhitungan operator sobel:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \cdot \dots \cdot 3$$

Berikut adalah contoh penggunaan operator sobel. Konvolusi pertama dilakukan terhadap *pixel* yang bernilai 1 (titik pusat mask). Perhitungan dilakukan dengan:

- 3 baris pertama dan 3 kolom pertama pada citra semula dikali S_x .
- 3 baris pertama dan 3 kolom pertama pada citra semula dikali S_y

$$\begin{bmatrix} 3 & 4 & 2 & 5 & 1 \\ 2 & 1 & 6 & 4 & 2 \\ 3 & 5 & 7 & 1 & 3 \\ 4 & 2 & 5 & 7 & 1 \\ 2 & 5 & 1 & 3 & 2 \end{bmatrix}$$

Gambar 9 CitraSemula

$$\begin{bmatrix} * & * & * & * & * \\ * & 18 & * & * & * \end{bmatrix}$$

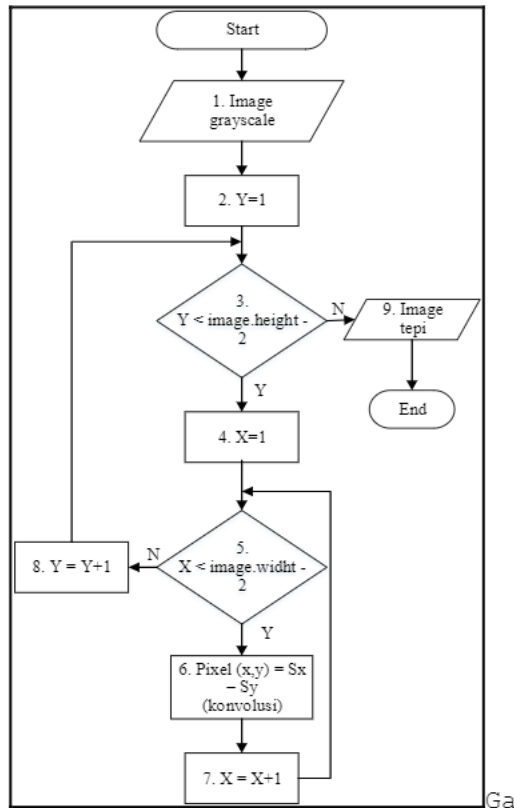
Gambar 10 Hasil Konvolusi

Nilai pada gambar 10 pada citra hasil konvolusi didapatkan dengan perhitungan :

$$S_x = (3)(-1) + (2)(-2) + (3)(-1) + (2)(1) + (6)(2) + (7)(1) = 11$$

$$S_y = (3)(1) + (4)(2) + (2)(1) + (3)(-1) + (5)(-2) + (7)(-1) = -7$$

$$S_x - S_y = 11 - (-7) = 18$$



Gambar 9 Proses Deteksi Tepi

Flowchart pada gambar 9 dijelaskan sebagai berikut:

1. Masukkan *imagegrayscale*.
2. $Y = 1$.
3. Apakah $Y < \text{image height} - 2$, jika ya lakukan langkah 5, jika tidak lakukan langkah 9.
4. $X = 1$.
5. Apakah $X < \text{image width} - 2$, jika ya lakukan langkah 7, jika tidak lakukan langkah 8.
6. $\text{Pixel}(x,y) = S_x - S_y$ (konvolusi).
7. $X = X + 1$, kembali ke langkah 6.
8. $Y = Y + 1$, kembali ke langkah 4.
9. Keluaran *image tepi*.

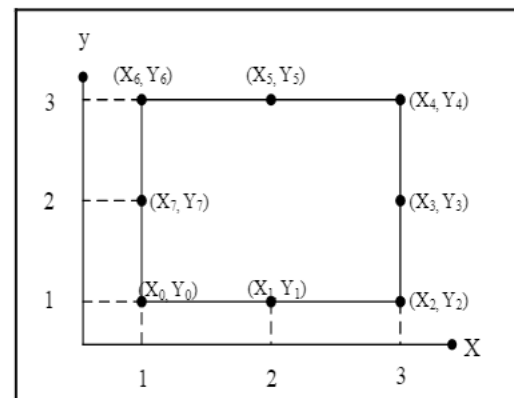
Proses Menghitung Pusat Massa Obyek (Center Of Gravity)

Center of Gravity adalah metode yang dipergunakan untuk menentukan titik keseimbangan dari grafik yang merupakan hasil dari proses deteksi tepi. Persamaan 5 merupakan perumusan matematis dari *center of gravity*.

$$\text{Posisi } X = \left(\frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \right) \dots \dots (5)$$

$$\text{Posisi } Y = \left(\frac{y_1 + y_2 + y_3 + \dots + y_n}{n} \right) \dots \dots (5)$$

Berikut contoh menghitung *Center Of Gravity*.

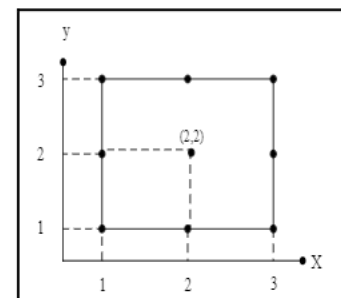


Gambar 8 Menghitung Center Of Gravity

$$\text{Posisi } X = \left(\frac{1 + 2 + 3 + 3 + 3 + 2 + 1 + 1}{8} \right) = 2$$

$$\text{Posisi } Y = \left(\frac{1 + 1 + 1 + 2 + 3 + 3 + 3 + 2}{8} \right) = 2$$

Jadi $(x,y) = (2,2)$



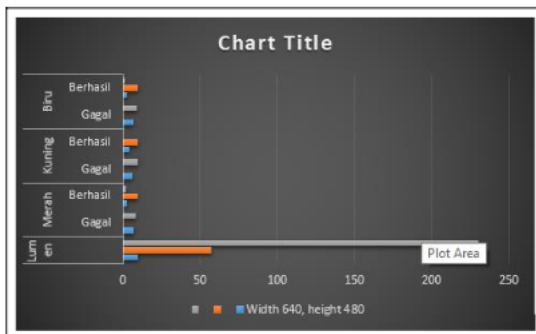
Gambar 9 Menghitung Center Of Gravity

Hasil Pengujian

Pengujian dilakukan terhadap 3 variable yaitu resolusi kamera, intensitas cahaya dan warna penanda jari tangan. Dalam *variable* resolusi kamera terdapat 4 resolusi yang telah diuji yaitu dengan ukuran *width* = 640, *height* = 480, *width* = 848, *height* = 480, *width* = 960, *height* = 540, dan *width* = 1280, *height* = 720. Untuk *variable* intensitas cahaya terdapat 3 parameter ukuran cahaya dalam satuan lumen yaitu cahaya rendah 10 lumen, cahaya sedang 57 lumen dan cahaya tinggi 230 lumen. *Variable* warna yang digunakan merupakan warna RGB yang terdiri dari merah, kuning, biru. Setiap resolusi kamera, intensitas cahaya dan warna telah dilakukan pengujian sebanyak 10 kali.

Tabel 1 Hasil Pengujian Resolusi *Width* 640
Height 480

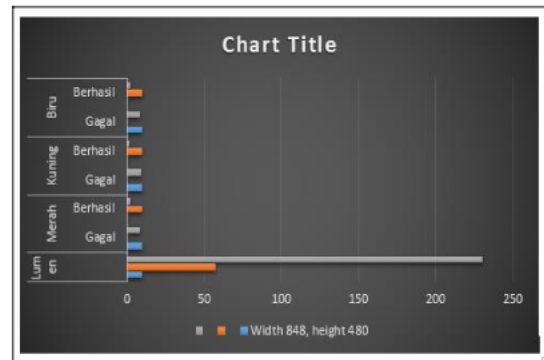
Resolusi	Lumen	Merah		Kuning		Biru	
		Gagal	Berhasil	Gagal	Berhasil	Gagal	Berhasil
Width 640, height 480	10	7	3	6	4	7	3
	57	0	10	0	10	0	10
	230	8	2	10	0	9	1



Gambar 10 Hasil Pengujian Resolusi *Width* 640
Height 480

Tabel 2 Hasil Pengujian Resolusi *Width* 848
Height 480

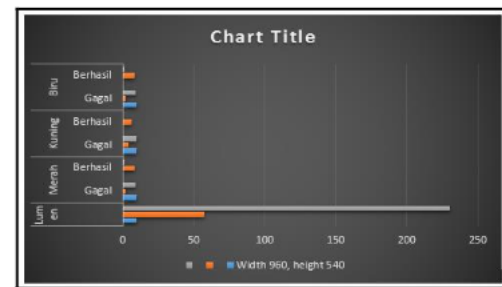
Resolusi	Lumen	Merah		Kuning		Biru	
		Gagal	Berhasil	Gagal	Berhasil	Gagal	Berhasil
Width 848, height 480	10	10	0	10	0	10	0
	57	0	10	0	10	0	10
	230	8	2	9	1	8	2



Gambar 11 Hasil Pengujian Resolusi *Width* 848
Height 480

Tabel 3 Hasil Pengujian Resolusi *Width* 960
Height 540

Resolusi	Lumen	Merah		Kuning		Biru	
		Gagal	Berhasil	Gagal	Berhasil	Gagal	Berhasil
Width 960, height 540	10	10	0	10	0	10	0
	57	2	8	4	6	2	8
	230	9	1	10	0	9	1



Gambar 12 Hasil Pengujian Resolusi *Width* 960
Height 540

Tabel 4 Hasil Pengujian Resolusi *Width* 1280
Height 720

Resolusi	Lumen	Merah		Kuning		Biru	
		Gagal	Berhasil	Gagal	Berhasil	Gagal	Berhasil
Width 1280, height 720	10	10	0	10	0	10	0
	57	4	6	5	5	3	7
	230	10	0	10	0	10	0



Gambar 13 Hasil Pengujian Resolusi Width 1280 Height 720

Hasil Keseluruhan Pengujian

Tingkat kesuksesan (%) untuk 10 kali pengujian terhadap setiap *variable* dapat dihitung dengan persamaan.

$$\frac{\text{Hasil Yang Sesuai/Benar}}{\text{Total Seluruh Uji Coba}} \times 100\% \dots 6$$

- Pada tabel 1 dapat disimpulkan tingkat keberhasilan sistem pada resolusi *width* = 640, *height* = 480 untuk intensitas cahaya 10 lumen terhadap warna merah 30%, warna kuning 40%, warna biru 30%. Untuk intensitas cahaya 57 lumen terhadap warna merah 100%, warna kuning 100%, warna biru 100%. Dan untuk intensitas cahaya 230 lumen terhadap warna merah 20%, warna kuning 0%, warna biru 10%.
- Pada tabel 2 dapat disimpulkan tingkat keberhasilan sistem pada resolusi *width* = 848, *height* = 480 untuk intensitas cahaya 10 lumen terhadap warna merah 0%, warna kuning 0%, warna biru 0%. Sedangkan untuk intensitas cahaya 57 lumen terhadap warna merah 100%, warna kuning 100%, warna biru 100%. Untuk intensitas cahaya 230 lumen terhadap warna merah 20%, warna kuning 10%, warna biru 20%.
- Pada tabel 3 dapat disimpulkan tingkat keberhasilan sistem pada resolusi *width* = 960, *height* = 540 untuk intensitas cahaya 10 lumen terhadap warna merah 0%, warna kuning 0%, warna biru 0%. Untuk intensitas cahaya 57 lumen terhadap warna merah 80%, warna kuning 60%, warna biru 80%. Dan untuk intensitas cahaya 230 lumen terhadap warna merah 10%, warna kuning 0%, warna biru 10%.

- Pada tabel 4 dapat disimpulkan tingkat keberhasilan sistem pada resolusi *width* = 1280, *height* = 720 untuk intensitas cahaya 10 lumen terhadap warna merah 0%, warna kuning 0%, warna biru 0%. Untuk intensitas cahaya 57 lumen terhadap warna merah 60%, warna kuning 50%, warna biru 70%. Dan untuk intensitas cahaya 230 lumen terhadap warna merah 0%, warna kuning 0%, warna biru 0%.

Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat diambil kesimpulan bahwa:

- Pengembangan aplikasi gitar *virtual* dengan penanda jari menggunakan *Color Filtering* telah berhasil dilakukan. Sistem mampu mendeteksi dan memfilter penanda warna jari.
- Proses *Color Filtering* dalam menangkap warna pada penanda jari dan memposisikan jari tepat dititik kord akan lebih baik agar sistem mengeluarkan suara kord gitar pada resolusi *Width* 640, *Height* 480 dan intensitas cahaya dengan lumen 57.

REFERENSI

1. _____. 2015 "Definisi Gitar". *Guitarxchord Wordpress*.
<https://guitarxchord.wordpress.com/guitar-2/definisi-gitar-sejarah-gitar/>
Dijadikan referensi untuk teori penelitian pada Oktober 2015.
2. Abdul Kadir, Adhi Susanto. (2013). "Teori dan Aplikasi Pengolahan Citra".
3. Benedictus Yoga Budi Putranto. (2010). "Segmentasi Warna Citra Dengan Deteksi HSV Untuk Mendeteksi Objek".
4. Havaluddin, 2011, "Memahami Penggunaan UML (Unified Modelling Language)".
5. I Made Santika Putra. 2014. "Segmentasi Citra Untuk Deteksi Objek Warna Pada Aplikasi Pengambilan Bentuk Citra *Rectangle*".
6. Wawan Kurniawan. 2011. "Pengenalan Bahasa Isyarat Dengan Menggunakan Metode Segmentasi Warna Kulit dan *Center Of Gravity*".

