

IMPLEMENTASI KALMAN FILTER DALAM TEKNIK HAND TRACKING SEBAGAI KONTROL POINTER MOUSE KOMPUTER

Yuza Reswan¹, Dedy Agung Prabowo²

¹Jurusan Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Bengkulu

²Jurusan Sistem Informasi, Fakultas Teknik, Universitas Muhammadiyah Bengkulu

Kampus 1 Jl. Bali Kota Bengkulu 38119 PO Box 118 Telp (0736)22765

¹ yuzareswan@umb.ac.id, ²dedyagungprabowo@umb.ac.id

ABSTRACT

In their daily life, humans usually use hands for the interaction between humans and machines or humans and human so that the human-computer interaction and computer-based Computer Vision using fingers has more value than using other objects. Many techniques are developed based on the Computer Vision model, one of the technique that is implemented in this field is Hand Tracking. The use of such techniques uses certain algorithms to improve performance, one of which is the filter algorithm. Kalman Filter is an estimator algorithm in which information from an object is detected in a frame and the status of the object in the previous frame to create an approximate status of the new state of the object so that the process of unstable object movements in the tracking process that occurs due to noise can be minimized by the kalman filter. The use of Hand Tracking techniques and Kalman Filter algorithm is expected to meet the needs of more advanced human and computer interactions. The results of this study are expected to be useful as a means of scientific development in the field of informatics engineering and the ability to build software in the field of Hand Tracking techniques and Kalman Filter algorithm as well as expected This application can be used for personal or public interest.

Keywords: Computer Vision, Hand Tracking, kalman Filter

ABSTRAK

Dalam kesehariannya, manusia seringkali menggunakan tangan untuk interaksi antara manusia dan mesin ataupun manusia dengan manusia sehingga interaksi manusia dan komputer berbasis Computer Vision menggunakan jari tangan memiliki nilai lebih dibandingkan menggunakan objek lain. Banyak teknik yang dikembangkan berdasarkan model Computer Vision, salah satu teknik yang sering diimplementasikan dalam bidang ini adalah Hand Tracking. Sering kali penggunaan teknik tersebut menggunakan algoritma tertentu untuk meningkatkan kinerjanya, salah satunya adalah algoritma kalman filter. Kalman Filter adalah sebuah algoritma estimator yang dimana informasi dari objek terdeteksi di suatu frame dan status objek tersebut pada frame sebelumnya untuk membuat status perkiraan dari status baru objek tersebut sehingga proses pergerakan objek yang tidak stabil pada proses tracking yang terjadi karena noise dapat diminimalkan oleh kalman filter. Penggunaan teknik Hand Tracking dan algoritma Kalman Filter diharapkan mampu memenuhi kebutuhan interaksi manusia dan komputer yang lebih maju. Hasil penelitian ini diharapkan berguna sebagai sarana pengembangan keilmuan dalam bidang teknik informatika dan kemampuan dalam membangun perangkat lunak pada bidang teknik Hand Tracking dan algoritma Kalman Filter serta diharapkan Aplikasi ini dapat digunakan untuk kepentingan pribadi maupun umum.

Kata Kunci : Computer Vision, Hand Tracking, kalman Filter

A. PENDAHULUAN

Computer Vision merupakan pengembangan dari pengolahan citra digital yang menambahkan kemampuan untuk dapat memahami dan mengerti sebuah citra. Telah sering dilakukannya penelitian dalam bidang ini diantaranya menggunakan mata manusia, gerakan tangan, gerak isyarat, gerak tubuh, gerak jari, dan sebagainya. Dalam kesehariannya manusia seringkali menggunakan tangan untuk interaksi antara manusia dan mesin ataupun manusia dengan manusia sehingga interaksi manusia dan komputer berbasis *Computer Vision*

menggunakan jari tangan memiliki nilai lebih dibandingkan menggunakan objek lain. Banyak teknik yang dikembangkan berdasarkan model *Computer Vision*, salah satu teknik yang sering diimplementasikan dalam bidang ini adalah *Hand Tracking*.

Tracking merupakan salah satu implementasi teknik *Tracking* dan *motion*, dimana *Tracking* dilakukan pada objek tangan manusia, objek tersebut merupakan hasil video dari webcam secara realtime, lalu dilakukan pendeteksian *motion* atau gerakan dari tangan pengguna tersebut, sehingga komputer dapat mendeteksi gerakan dan arah tangan dari pengguna tersebut yang terlihat pada webcam.

B. LANDASAN TEORI

1. *Computer Vision*

Computer Vision adalah salah satu teknologi yang paling banyak dipakai pada zaman ini. Teknologi *Computer Vision* ini merupakan salah satu bidang dari teknologi *Artificial Intelligence*. *Computer Vision* juga merupakan kumpulan dari metode-metode untuk mendapatkan, memproses, menganalisis suatu gambar atau dalam arti lain *Computer Vision*, merupakan kumpulan metode-metode yang digunakan untuk menghasilkan angka-angka atau simbol-simbol yang didapat dari gambar yang diambil dari dunia nyata agar komputer dapat mengerti apa makna dari gambar tersebut. Inti dari teknologi *Computer Vision* adalah untuk menduplikasi kemampuan penglihatan manusia kedalam benda elektronik sehingga benda elektronik dapat memahami dan mengerti arti dari gambar yang dimasukkan. [1]

Computer Vision dapat dikatakan sebagai transformasi atau perubahan dari data-data yang dapat berupa gambar diam ataupun video kamera menjadi bentuk lain atau suatu representasi baru dan membantu dalam pengambilan keputusan. [2]

Semua bentuk transformasi yang dilakukan di arsipkan atau disimpan untuk tujuan tertentu. Input data dapat berasal dari informasi yang berhubungan dengan objek. Pada representasinya baru dimungkinkan berarti merubah warna dari suatu citra menjadi citra yang *grayscale* atau menghilangkan pergerakan kamera dari suatu citra berurutan.

Dalam disiplin ilmu, *Computer Vision* berkaitan dengan teori-teori dibalik sistem buatan yang mengekstrak informasi dari gambar. Informasi yang diekstrak dari gambar dapat berupa data-data yang berbeda-beda, seperti urutan jalannya video, intensitas cahaya, atau perspektif dari sudut gambar yang berbeda-beda.

Dalam disiplin teknologi, *Computer Vision* mengusahakan cara agar teori-teori dan model dapat diterapkan untuk pembangunan sistem pada sistem komputer. Contoh beberapa aplikasi yang menggunakan *Computer Vision* seperti, alat navigasi, kontroler, *sensor tracking*, *face detection*, serta *pose detection*.

2. Pengolahan Citra

Pengolahan citra adalah salah satu cabang dari ilmu informatika. Pengolahan citra berkuat pada usaha untuk melakukan transformasi suatu citra/gambar menjadi citra lain dengan menggunakan teknik tertentu. Pengolahan citra merupakan bidang yang bersifat multi-disiplin, yang terdiri dari banyak aspek, antara lain fisika, elektronika, matematika, seni dan teknologi komputer. Pengolahan citra (*image processing*) memiliki hubungan yang sangat erat dengan disiplin ilmu yang lain. Input dari pengolahan citra adalah citra, sedangkan outputnya adalah citra hasil pengolahan. [3].

3. Tracking

Tracking merupakan istilah untuk melacak atau mencari suatu objek pada suatu citra, dalam prosesnya ada sesuatu yang penting yang terjadi pada objek tersebut dalam citra, ketika citra tersebut merupakan *frame* yang merupakan subquense dari suatu video yaitu *motion*, *motion Tracking* adalah proses dimana visual suatu *image* terdeteksi. [4]

4. Motion Detection

Motion merupakan pergerakan atau arah pergerakan yang terlihat dilakukan objek atau yang terjadi terhadap objek akibat pergantian *frame*. Deteksi gerakan (*Motion Detection*) adalah proses mendeteksi perubahan posisi dari suatu obyek relatif terhadap sekitarnya atau perubahan lingkungan relatif terhadap suatu objek. Algoritma deteksi gerakan dimulai dengan bagian segmentasi dimana objek yang bergerak telah tersegmentasi dari latar belakangnya. [5]

Motion detection dapat dicapai oleh kedua metode mekanik dan elektronik. Ketika mendeteksi gerakan dilakukan dengan organisme alami, hal itu disebut persepsi gerak. Gerakan dapat dideteksi oleh:

- a. Infrared (sensor pasif dan aktif)
- b. Optik (video dan kamera sistem)
- c. Energi Frekuensi Radio (radar, microwave dan deteksi gerakan tomografi)
- d. Suara (mikrofon dan sensor akustik)
- e. Getaran (triboelectric, seismik, dan sensor inersia-switch)
- f. Magnit (sensor magnetik dan magnetometer)

5. Threshold

Dalam dunia pengolahan citra *Threshold* merupakan bagian *image processing* (pengolahan citra) dimana prosesnya melakukan pengolahan piksel pada suatu citra atau menghilangkan beberapa piksel dan juga mempertahankan beberapa piksel sehingga menghasilkan suatu citra baru hasil sortir piksel yang telah dilakukan. Dengan dilakukannya komputasi threshold maka dapat dengan mudah mendapatkan *edge*/tepi dari suatu citra. *Thresholding* adalah metode *image segmentation* yang populer dikarenakan

metode ini adalah metode tersimpel dan mudah di implementasikan, hal ini berguna dalam membedakan latar depan dari latar belakang. [6]

Properti sederhana dapat piksel bagi di suatu daerah adalah intensitas. Jadi, cara alami untuk mensegmen daerah tersebut melalui proses *thresholding*, Dari sebuah gambar *grayscale*, *thresholding* dapat digunakan untuk menciptakan gambar *binary* [7].

6. Gaussian Filter

Gaussian filter adalah suatu filter dengan nilai pembobotan pada setiap piksel dipilih berdasarkan bentuk fungsi Gaussian. Filter ini sangat baik dan sering digunakan untuk menghilangkan *noise* yang bersifat sebaran normal, yang banyak dijumpai pada citra hasil proses digitalisasi menggunakan kamera, hal ini terjadi karena merupakan fenomena alamiah akibat dari sifat pantulan cahaya dan kepekaan sensor cahaya pada kamera itu sendiri. [8]. Persamaan distribusi Gaussian 2D adalah sebagai berikut:

$$G(x, y) = \frac{1}{S\sqrt{\pi}} e^{-((x-m_x)^2+(y-m_y)^2)}$$

Dimana : $G(x,y)$: adalah gaussian dua dimensi.

s : adalah sebaran dari fungsi gaussian.

(m_x, m_y) : adalah titik tengah dari fungsi gaussian.

Bedasarkan rumus dari fungsi Gaussian diatas untuk ukuran 3x3 diperoleh matrik kernel filter Gaussian :

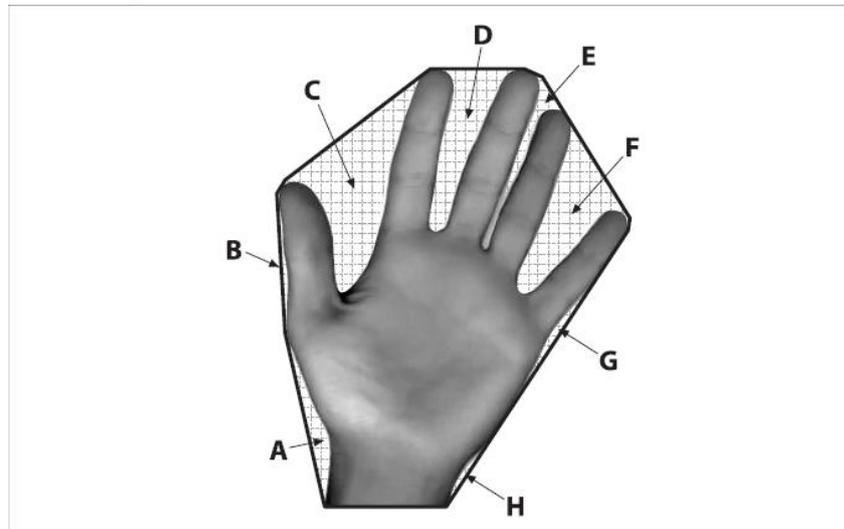
$$H = \begin{Bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{Bmatrix} / 13 \rightarrow \text{atau} \rightarrow H = \begin{Bmatrix} 0.077 & 0.077 & 0.077 \\ 0.077 & 0.308 & 0.077 \\ 0.077 & 0.077 & 0.077 \end{Bmatrix}$$

7. Convexity dan Convexity Defects

Dalam menggambar *shape* atau suatu bentuk dalam suatu kontur yang sudah ditemukan dapat menggunakan metode *Convexity* dan *Convexity defects*. Metode ini memberikan pemahaman pada komputer mengenai bentuk dari kontur yang ditemukan dengan memanfaatkan ketidaktepatan penggambaran *shape* yang mengelilingi objek, metode ini bukan hanya dapat mendapatkan karakteristik dari tangan tapi juga posisi tangan.

Untuk menemukan ujung jari dapat dilakukan dengan dua langkah. Yaitu yang pertama menentukan kontur tangan. Yang kedua menemukan *convex* dan *defects*-nya. *Convex hull* merupakan cara untuk memberikan titik awal dan akhir dari suatu puncak atau kelengkungan dari telapak tangan. Setelah mendapatkan nilai atau poin-poin maka kita

dapat menentukan nilai kecuramannya atau kedalaman antar lengkungan. *Defects* merupakan segmentasi yang terbentuk dari dua lengkungan yang telah dideteksi oleh convex hull. Hal ini tergambar pada gambar dibawah dimana A diasumsikan sebagai titik awal convex dan B sebagai titik akhir convex.



Gambar 1. Garis berkontur gelap adalah convex hull sekitar tangan daerah grid (A-H) adalah convexity defects. [9]

Convexity Defects berfungsi sebagai suatu metode untuk mendapatkan kontur yang berkarakter kuat dan pasti pada suatu objek misalkan tangan. Sedangkan convex hull biasa dipakai untuk mendapatkan kontur besar yang mengelilingi suatu objek.

Convexity dan *convexity defects* menggambarkan kontur dengan dua metode yaitu convex hull dan *convexity defects*. *Convexity* adalah rasio perimeter *convexity-hull*, $\pi_{convex\ hull}$ ke perimeter batas bentuk, π_{shape} , dimana *convex hull* adalah poligon convex minimum yang mencakup bentuk.

$$\zeta = \frac{\pi_{convex\ hull}}{\pi_{shape}}$$

Sehingga dengan menggunakan convex hull kita akan mendapatkan bentuk boundary dari tangan dengan batas-batas minimal. Sedangkan untuk mendapatkan kontur atau menggambarkan kontur tepat mengelilingi kontur tangan dapat menggunakan metode *convexity defects*. *Convexity Defects* memberikan suatu sarana pengkarateristikan tidak hanya tangan itu sendiri tetapi juga bagian-again dari tangan tersebut. [9]

8. Kalman Filter

Algoritma Kalman Filter dikembangkan oleh Rudolf E. Kalman pada tahun 1960, *Kalman Filter* (KF) merupakan suatu metode estimasi variabel keadaan dari sistem dinamik stokastik linear diskrit yang meminimumkan kovarian error estimasi

$$x_{k+1} = A_k x_k + B_k u_k + ,$$

$$z_k = H_k x_k + v_k,$$
$$x_0 \sim N(x_0, P_0); w_k \sim N(0, Q_k); v_k \sim N(0, R_k).$$

dengan:

x_0	= inisial dari sistem
x_{k+1}	= variabel keadaan pada waktu $k + 1$ dan berdimensi $n \times 1$
x_k	= variabel keadaan pada waktu k yang nilai estimasi awalnya x_0 dan kovarian awal P_0 , $x_k \in R^n$
u_k	= vektor masukan deterministik pada waktu k , $u_k \in R^n$
w_k	= <i>noise</i> pada sistem dengan mean $w_k = 0$ dan kovarian Q_k
z_k	= variabel pengukuran, $z_k \in R^m$
v_k	= <i>noise</i> pada pengukuran dengan mean $v_k = 0$ kovarian R_k
$A_k B_k H_k$	= matriks-matriks dengan nilai elemen-elemennya adalah koefisien variabel masing-masing.

Pada KF, estimasi dilakukan dengan dua tahapan, yaitu tahap prediksi (*time update*) dan tahap koreksi (*measurement update*). Tahap prediksi merupakan tahap untuk memprediksi variabel keadaan berdasarkan sistem dinamik dengan menggunakan persamaan estimasi variabel keadaan. Tingkat ketelitian pada tahap ini dihitung menggunakan persamaan kovarian *error*.

Sedangkan pada tahap koreksi, hasil estimasi variabel keadaan yang diperoleh pada tahap prediksi dikoreksi menggunakan model pengukuran. Tingkat ketelitiannya dihitung menggunakan persamaan kovarian *error* yang melibatkan matriks *Kalman Gain*. Matriks ini digunakan untuk meminimumkan nilai kovarian *error* [10]

C. HASIL PENELITIAN

1. Butir Pengujian Hasil Tangkapan Webcam

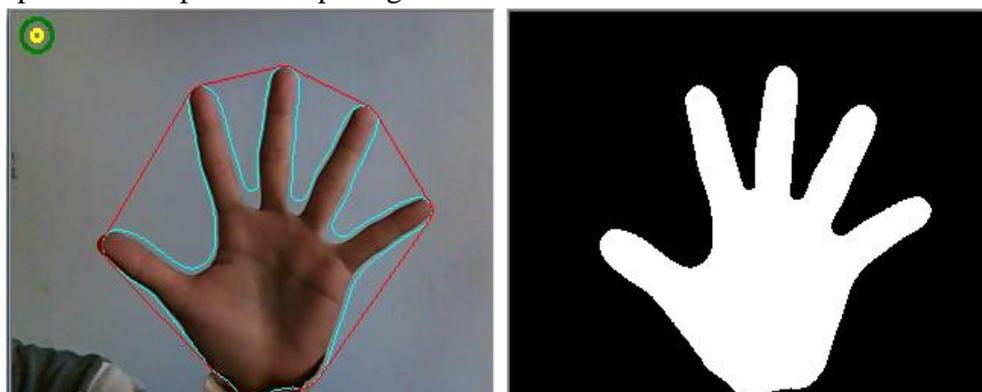
Pada aplikasi ini sistem dimulai dengan menginputkan citra dari hasil tangkapan kamera webcam yang kemudian yang kemudian sistem dapat mengenali input dari kamera webcam tersebut. Implementasi penangkapan serta menampilkan hasil tangkapan webcam menggunakan fungsi *QueryFrame()* yang dimana hasilnya dapat dilihat pada gambar 2



Gambar 2. Hasil Tangkapan Kamera Webcam

2. *Butir Pengujian Proses Segmentasi Warna Kulit*

Citra yang telah ditangkap melalui kamera webcam kemudian dilakukan proses segmentasi yang bertujuan untuk memisahkan warna kulit yang telah ditentukan parameternya pada jenis warna YcrCb yang bernilai minimum (0, 131, 80) dan bernilai maksimum (255, 185, 135) sehingga mengklasifikasi warna kulit dari background. Hasil pada proses ini dapat dilihat pada gambar 3 berikut.



Gambar 3. Hasil Segmentasi

3. *Butir Pengujian Proses Filter Gaussian*

Dengan menggunakan fungsi Gaussian, Citra yang telah diklasifikasikan sesuai dengan warna tangan dapat difilter sisi-sisi dari citra yang terdeteksi sebagai warna kulit dan bukan warna kulit sehingga hasil citra dari klasifikasi tersebut didapatkan kontur yang

lebih halus. Gambar 4 dibawah memperlihatkan perbedaan citra yang tanpa menggunakan filter gaussian dan yang menggunakan filter gaussian.



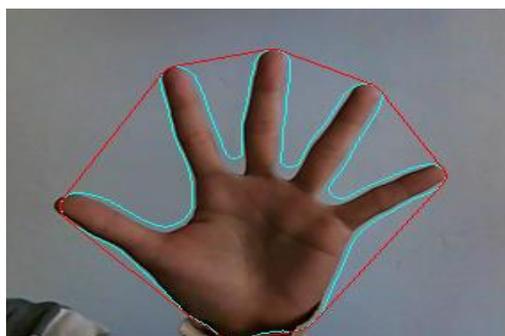
(a)

(b)

Gambar 4. (a)Citra Tanpa *Gaussian filter*, (b)Citra Dengan *Gaussian Filter*

4. *Butir Pengujian Proses Convexity Hull dan Defect*

Pada aplikasi ini proses Convexity Hull dan Defect berfungsi sebagai metode yang dapat menentukan kontur tangan serta jadi hasil dari segmentasi yang membutuhkan citra dengan nilai warna 2 bit. Dengan fungsi Convexity Hull dan Defect pada Microsoft Visual Studio 2010 C# disertai dengan library EmguCV sistem melakukan pencarian kontur tangan setelah proses segmentasi terdeteksi yang hasilnya dapat dilihat pada gambar 5.

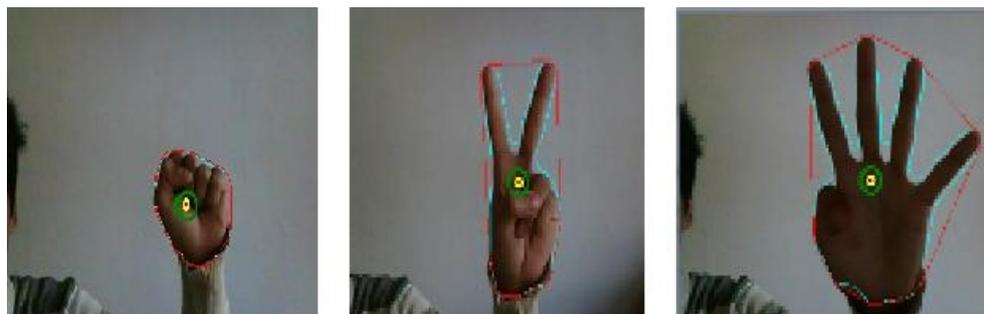


Gambar 5. Convexity Hull dan Defect

Garis berwarna merah pada gambar 5 adalah batas himpunan titik luar(Hull) yang berarti kontur terbesar dari citra tangan yang telah tersegmentasi dan garis berwarna cyan adalah batas himpunan titik luar(Hull) yang memiliki titik celah yang diartikan titik *defect* tertinggi celah adalah ujung jari dan titik *defect* terendah adalah bagian di dalam diantara dua jari.

5. *Butir Pengujian Proses Komputasi Jari*

Hasil pada pencarian kontur tangan pada proses Convexity Hull dan Defect dilanjutkan dengan proses komputasi jari yang bertujuan menghitung jari yang terdeteksi pada citra yang nantinya digunakan sebagai kontrol dalam melakukan klik kanan ataupun kiri pada sistem. Pada gambar 6 menjelaskan bahwa ketika tangan terdeteksi dan melakukan proses Convexity Hull dan Defect nilai array dari fungsi *defect* yang berisi titik kordinat jari tersebut didefinisikan sebagai jari sehingga komputasi jari dimulai



Gambar 6. Komputasi Jari

6. *Butir Pengujian Pencarian Titik Tengah Kontur*

Dalam prosesnya sistem ini membutuhkan titik tengah kontur sebagai acuan kordinat pointer pada layar. Pencarian titik tengah kontur dan posisi kordinatnya dilakukan setelah didapatkan kontur terbesar dari hull dengan menjalankan fungsi *cvMoments* pada penulisan kode sehingga didapatkan titik tengah yang ditandai dengan lingkaran berwarna kuning pada gambar 7



Gambar 7. Titik Tengah Kontur Tangan

7. *Butir Pengujian Kalman Filter*

Setelah didatakannya titik tengah kontur yang dimaksudkan sebagai penentu posisi kordinat pointer mouse pada layar didalam penelitian bidang pengolahan citra terutama

pada penangkapan citra secara real-time menggunakan kamera webcam terdapat beberapa faktor yang mengganggu proses pengolahan citra seperti *noise* dan *framerate* kamera webcam sehingga diperlukannya suatu metode yang dapat mengurangi faktor-faktor tersebut. Dalam hal ini penulis menggunakan algoritma kalman filter dalam perancangan sistem ini. Lingkaran berwarna kuning Pada gambar 7 menunjukkan titik tengah kontur sedangkan lingkaran berwarna hijau adalah hasil dari proses filtering algoritma kalman terhadap titik tengah kontur yang dimana titik kalman tersebut disinkronisasikan pada pointer untuk menentukan posisi pointer pada layar.

Dalam implementasi algoritma kalman pada aplikasi ini terdapat state sederhana yang berisi 4 variabel dinamik yaitu kordinat x dan y serta percepatan x dan y pada variabel titik tengah kontur tangan dan 2 variabel pengukuran. 4 variabel dinamik tersebut mengisi matrik 4*4 berikut.

$$\{1, 0, 1, 0\}$$

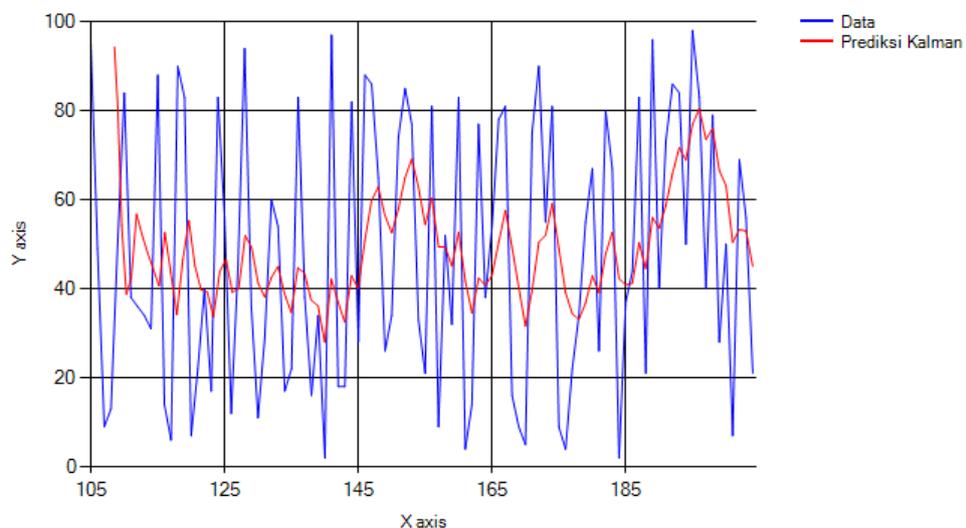
$$\{0, 1, 0, 1\}$$

$$\{0, 0, 1, 0\}$$

$$\{0, 0, 0, 1\}$$

4 variabel tersebut beserta gangguan *noisenya* diinput dalam matrik kalman yang selanjutnya akan di ukur pada proses kalman sehingga menghasilkan state yang baru yang dinyatakan sebagai prediksi kalman dan merupakan koordinat x,y baru yang telah dikurangi *noisenya* pada posisi pointer.

Berikut ini adalah contoh grafik pemrosesan filtering algoritma kalman.

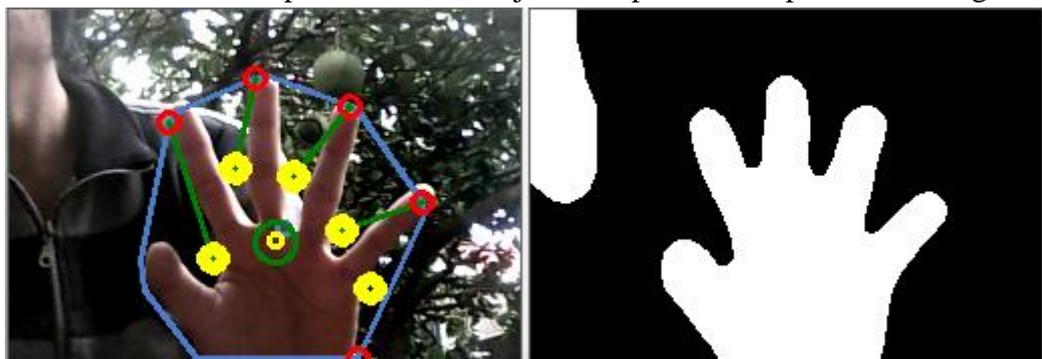


Gambar 8. Grafik Prediksi Kalman dan Data Awal

Dari grafik data yang diinputkan adalah data random yang dimana berisi nilai dari 4 variabel yang telah disebutkan, pada grafik tersebut terlihat jelas data masukan awal yang ditandai dengan warna biru menghasilkan keluaran yang tidak stabil dikarenakan faktor-faktor tertentu dan data prediksi kalman yang ditandai dengan warna merah adalah hasil pemrosesan filtering algoritma kalman terhadap data awal yang menghasilkan grafik data yang jauh stabil dan telah mengurangi gangguan *noise* yang dibawa dari data awal sebelumnya.

8. *Butir Pengujian Pengaruh Latar Belakang (Background)*

Pada penelitian ini telah disebutkan bahwa sistem mendeteksi tangan dengan cara memisahkan warna kulit tangan dengan latar lainnya sehingga pada butir 8 ini diuji pengaruh latar belakang ataupun lingkupan area tangkapan kamera terhadap sistem, pada gambar 9 dibawah menampilkan hasil dari uji sistem pada beberapa warna background.



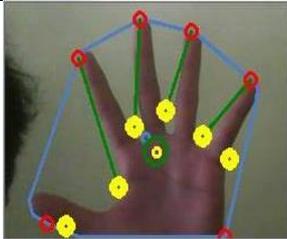
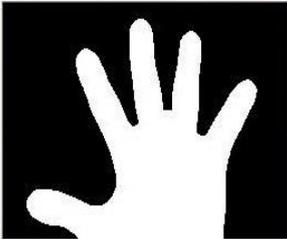
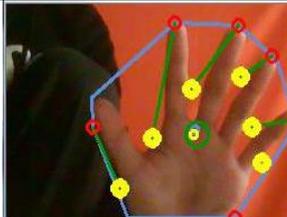
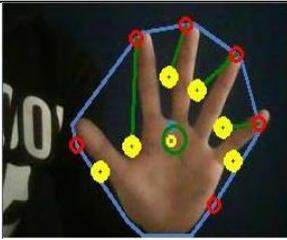
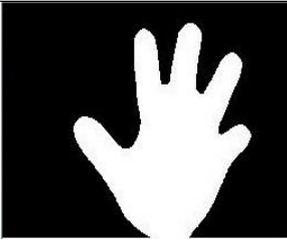
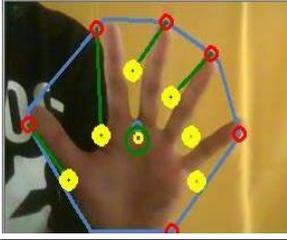
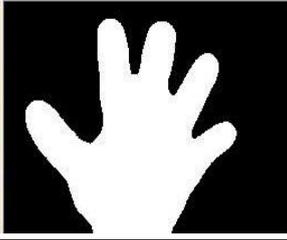
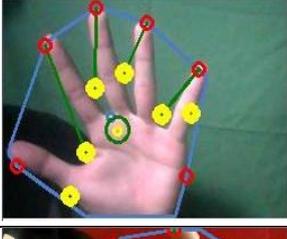
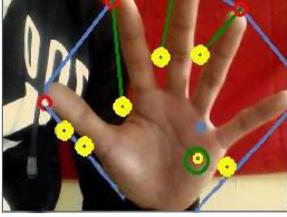
Gambar 9. Penentuan Kontur Tangan Sukses dengan Warna Latar Belakang Bervariasi

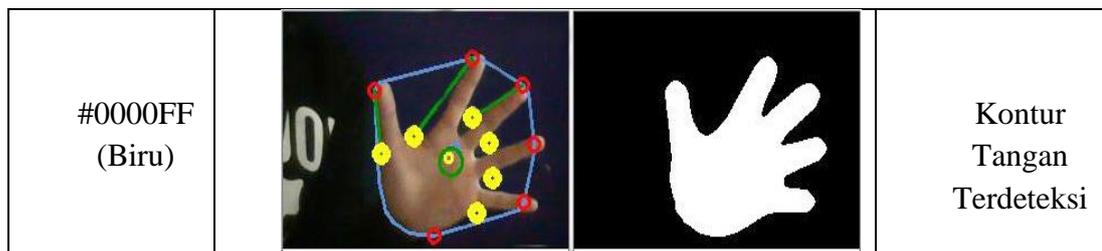
Terlihat bahwa range warna Ycbr pada kulit yaitu (0, 131, 80) pada ambang batas minimal dan (245, 185, 135) pada ambang batas maksimal sehingga warna selain pada nilai ambang batas tersebut tidak terdeteksi atau tidak dapat dinyatakan sebagai tangan.

Dibawah ini merupakan tabel dari hasil uji coba yang telah dilakukan untuk mengetahui pengaruh latar belakang pada beberapa warna:

Tabel 1. Pengujian Pengaruh Warna Latar Belakang

Kode Warna	Tampilan	Keterangan
------------	----------	------------

#F5F5DC (abu-abu)			Kontur Tangan Terdeteksi
#FFFF00 (Jingga)			Kontur Tangan Terdeteksi
#9400D3 (Ungu)			Kontur Tangan Terdeteksi
#FFFF00 (Kuning)			Kontur Tangan Terdeteksi
#008000 (Hijau)			Kontur Tangan Terdeteksi
#FF0000 (Merah)			Kontur Tangan Terdeteksi



D. KESIMPULAN

Berdasarkan hasil pengujian sistem yang telah dibuat ini diperoleh beberapa kesimpulan sebagai berikut :

1. Dari hasil penelitian yang dilakukan dapat terlihat aplikasi berjalan dengan cukup baik dan algoritma *confexity defect* dalam komputasi jari dan algoritma *kalman filter* untuk menghaluskan pergerakan mouse yang diimplementasikan bekerja sesuai dengan fungsinya. Namun kelemahan terjadi pada proses pendeteksian tangan, dimana gambar yang tertangkap webcam terdapat objek lain selain tangan berwarna sama yang menghasilkan ketidakstabilan pendeteksian dikarenakan pendeteksian warna berdasarkan warna kulit tangan sehingga apabila ada objek yang bersentuhan dengan tangan dengan warna yang sejenis dalam ruang lingkup tangkapan kamera webcam maka objek tersebut ikut terdeteksi menjadi bagian dari kontur tangan.
2. kualitas kamera yang beresolusi hanya 1.3 mp menghasilkan *framerate* yang rendah sehingga munculnya banyak *noise* yang mengganggu proses pendeteksian. Kualitas kamera menjadi faktor penting yang mempengaruhi kualitas hasil dari proses yang terjadi dalam sistem pendeteksian ini, semakin tinggi kualitas resolusi kamera maka semakin mampu mengurangi *noise* yang dapat mengganggu proses pendeteksian.
3. Selain kamera faktor selanjutnya adalah cahaya, cahaya juga salah satu faktor yang mempengaruhi kualitas hasil dari proses yang dilakukan oleh sistem. Semakin baik pencahayaan maka proses pendeteksian akan semakin baik.

DAFTAR PUSTAKA

- M. Sonka, V. Hlavac and R. Boyle, Image Processing, Analysis, and Machine
1] Vision, Nelson Education Limited, 2008.
Fadlisyah, Computer Vision dan Pengolahan Citra, Andi Publisher, 2007.

2]

T. Sutoyo, E. Mulyanto, V. Suhartono, O. D. Nurhayati and Wijanarto, Teori

3] pengolahan citra digital, Yogyakarta: Andi Publisher, 2009.

B. M. A. Wiradani, "Motion Tracking System Using Computer Vision With

4] OpenCV Source Software," 2010.

A. K. Sahu and A. Chouhey, "Motion Detection Surveillance System Using

5] Background Subtraction Algorithm," 2013.

A. K. Mandal and D. K. Baruah, "Image Segmentation Using Local

6] Thresholding And Ycbcr Color Space," 2013.

L. G. Shapiro and G. C. Stockman, Computer Vision, Prentice Hall, 2001.

7]

U. Ahmad, Pengolahan Citra Digital & Teknik Pemrogramannya, Graha Ilmu

8] Yogyakarta, 2005.

G. Bradski and A. Kaehler, Learning OpenCV, United States of America:

9] O'Reilly Media, 2008.

G. Welch and G. Bishop, "An Introduction to the Kalman Filter," 2006.

10]

Departemen Pendidikan Nasional, KAMUS BAHASA INDONESIA (Edisi

11] Keempat), Jakarta: Balai Pustaka, 2008.

Ramsoft, "Methodology," 2012. [Online]. Available:

12] <http://www.ramsoft.com.au/methodology.php>. [Accessed 06 2014].

D. H. Ballard and C. M. Brown, Computer Vision, New York: Prentice Hall,

13] 1982.