PENGENALAN POLA ISYARAT TANGAN PADA INPUT HAND GESTURE DINAMIS
CHRISTOPHEL.A.FINA

[1]Christophel. A. Fina, [2] Heani Budiyati, S.Si, M.Kom [3] Ag Rudatyo.H S.Si, M.Kom
[1,2,3]Jurusan Teknik Informatika, Fakultas Sains dan Komputer, Universitas Kristen Immanuel
Jalan Solo Km. 11 PO Box 4 YKAP Yogyakarta, ph: (0274) 496256-296247 fax: (0274) 496258
e-mail: *[1] chrissean99@gmail.com, *[2] heni@ukrimuniversity.ac.id, *[3] rudatyo@university.ac.id

Abstraksi

Penggunaan pose tangan merupakan salah satu solusi untuk memenuhi kebutuhan manusia dalam interaksi manusia dan komputer (Computer Human Interaction) yang lebih intuitif, cepat, dan sejalan dengan fungsi natural anggota tubuh manusia. Tangan dapat menghasilkan beragam pose untuk mengontrol suatu kondisii. Hal inilah yang mendorong penulis untuk mengangkat topik permasalahan ini menjadi penelitian dalam tugas akhir ini mengenai pengenalan isyarat tangan. Peneliti akan lebih fokus pada input dinamis dalam ruang lingkup pada pendekteksian ujung-ujung jari dari hasil segmentasi tangan.

Variabel yang merupakan tahapan pemrosesan pada program aplikasi sebagai perangkat dalam penelitian terdiri dari segmentasi pada obyek tangan sebagai foreground. Tahap berikutnya adalah deteksi kontur obyek tangan untuk proses menghitung jumlah titik tertinggi yang menentukan kondisi isyarat tangan yang terbuka dan tertutup.

Pose tangan frontal dan berhadap-hadapan pada lock area tidak tepat menghasilkan segmentasi obyek isyarat tangan dengan pose tangan terbalik, dari samping kiri dan kanan, dengan dua tangan dan pose tangan tidak frontal menghadap miring pada kamera. Sedangkan pose tangan teoat menghasilkan segmen obyek isyarat tangan dengan pose frontal (berhadap-hadapan) tegak lurus dan terbalik menghadap kamera. Pengenalan isyarat tangan dapat mengenali pose jari terbuka dan tidak mampu menghitung ujung jari yang terbuka dalam pose jari yang rapat. Pengenalan isyarat tangan efektif dengan pose jari terbuka dengan isyarat jari dua, tiga, empat dan lima jari sedangkan pose satu jari dikenali dalam isyarat tangan tertutup. Pengenalan isyarat tangan yang tepat dalam segmentassi tangan dengan jari tidak rapat telah berhasil mengenali pose tangan terbuka dan tertutup.

Kata kunci : segmentasi, binerisasi, deteksi ujung jari, convex hull.

.

Abstraction

The use of hand poses is one solution to meet human needs in human and computer interactions (Computer Human Interaction) that are more intuitive, fast, and in line with the natural functions of the human body members. Hands can produce a variety of poses to control a condition. This has encouraged the writer to raise the topic of this problem into research in this

thesis regarding the recognition of hand signals. Researchers will focus more on dynamic input within the scope of the finger-tip detection of the results of hand segmentation.
Variables that are the processing stages of the application program as a device in research consist of segmentation of hand objects as foreground. The next stage is the detection of the contours of the hand object for the process of counting the highest number of points that determine the condition of open and closed hand signals.
Frontal hand pose and face to face in the lock area does not produce the right segmentation of hand gesture objects with inverted hand poses, from the left and right sides, with two hands and not frontal hand poses facing tilted on the camera. Whereas the teoat hand pose produces a hand signal object segment with a frontal pose (facing each other) perpendicular and inverted facing the camera. Hand gesture recognition can recognize open finger poses and be unable to count open fingertips in tight finger poses. Hand gesture recognition is effective with open finger poses with two, three, four and five finger gestures while one finger poses are recognized in closed hand gestures. Recognition of proper hand signals in the hand segment with the fingers not tight has succeeded in recognizing open and closed hand poses.

Keywords: segmentation, binarization, fingertip detection, convex hull.

1. PENDAHULUAN

Penggunaan pose tangan merupakan salah satu solusi untuk memenuhi kebutuhan manusia dalam interaksi manusia dan komputer (*Computer Human Interaction*) yang lebih intuitif, cepat, dan sejalan dengan fungsi natural anggota tubuh manusia. Tangan dapat menghasilkan beragam pose. Setiap pose memiliki maksud dan makna tersendiri sesuai dengan kesepakatan umum ataupun kesepakatan diantara yang melakukan komunikasi. Oleh karena itu pose tangan dapat digunakan untuk menggantikan fungsi mouse, untuk mengontrol gerak robot, dan sebagainya Pengguna dapat melakukan gerakan sederhana untuk mengontrol atau berinteraksi tanpa menyentuh fisik pada perangkat. Banyak pendekatan telah dilakukan dengan menggunakan kamera dan algoritma visi komputer untuk menafsirkan isyarat dari badan dan tangan.

Salah satu bagian dari *gesture recognition* adalah *hand gesture recognition* yang merupakan pendeteksian pose tangan. Pengenalan hand gesture menjadi penelitian yang menarik untuk dikembangkan dalam *computer vision*. Pendeteksian pada ujung jari dari ragam pose tangan akan menjawab kebutuhan manusia dalam interaksi manusia.

Hal inilah yang mendorong penulis untuk mengangkat topik permasalahan ini menjadi penelitian dalam tugas akhir ini mengenai pengenalan isyarat tangan. Peneliti akan lebih fokus pada input dinamis dalam ruang lingkup pada pendekteksian ujung-ujung jari dari hasil segmentasi tangan pada area yang sudah ditentukan untuk mengenali isyarat tangan sebagai tangan terbuka atau tertutup dengan judul “Pengenalan Isyarat Tangan Pada Input Hand Gesture Dinamis”.

2. MATERI DAN METODE PENELITIAN

2.1. Sistem penglihat Komputer

Sistem penglihat komputer (*computer vision*) bertujuan untuk mengkomputerisasi penglihatan manuisa atau dengan kata lain membuat citra digital dari citra sebenarnya (sesuai dengan penglihatan manusia). Hal tersebut dapat disimpulkan input dari visi komputer adalah berupa citra penglihatan manusia sedangkan outputnya berupa citra digital.[1] Sistem penglihat komputer otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akusisi citra, pengolahan citra, pengenalan dan pembuatan keputusan. *Computer vision* mencoba meniru cara kerja sistem visual manusia (*human vision*) yang sesungguhnya sangat kompleks, bagaimana manusia melihat objek dengan indra penglihatan (mata), lalu citra objek diteruskan ke otak untuk diinterpretasi sehingga manusia mengerti objek apa yang tampak dalam pandangan mata.. (Darmawan, 2009). Informasi dari suatu gambar yang diambil dari berbagai banyak bentuk, seperti urutan video, pandangan dari beberapa kamera, atau data multi-dimensi dari scanner medis dapat diperlukan untuk menyelesaikan tugas tertentu. *Computer vision* berusaha untuk menerapkan teori dan model untuk pembangunan sistem *computer vision*. *Computer vision* lebih cenderung yang mempelajari bagaimana komputer dapat mengenali obyek yang diamati.

2.2. Pengolahan Citra

Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Teknik pengolahan citra menerapkan standar-teknik pemrosesan sinyal digunakan untuk optik dan pengolahan gambar analog. Akuisisi gambar atau yang menghasilkan gambar input di tempat pertama disebut sebagai pencitraan. Penerapan proses pengolahan citra adalah setiap bentuk pengolahan sinyal dimana input adalah gambar, seperti foto atau video bingkai, sedangkan output dari pengolahan gambar dapat berupa gambar atau sejumlah karakteristik atau parameter yang berkaitan dengan gambar.

2.2.1. Binerisasi (Metode *Ni-Black*)

Binerisasi adalah langkah awal dari kebanyakan proses dalam sistem analisis dokumen citra dan mengacu pada konversi dari gambar *grayscale* menjadi gambar biner. Dalam literatur, binerisasi biasanya dilaporkan dilakukan baik secara global maupun lokal. Metode global (global *thresholding*) menggunakan nilai ambang tunggal untuk mengklasifikasikan piksel gambar ke kelas objek atau latar belakang sedangkan skema lokal (*adaptif thresholding*) dapat menggunakan beberapa nilai yang dipilih sesuai dengan informasi daerah setempat. Metode *Ni-Black* merupakan binerisasi untuk segmentasi citra dengan *tresholding* (pengambangan) untuk mengelompokan *foreground* dan *background*. *Ni-Black* berfungsi untuk menghitung rata rata piksel dan standar deviasi dalam r x r beberapa piksel yang saling bertetanggaan, kemudian citra disegmentasikan menggunakan persamaan sebagai berikut.

---

[1] Darma Putra. *Pengolahan Citra Digital*. (Andi Offset : 2010 Yogyakarta). Hal 2.

$$T(x,y) = \mu(x,y) + k * s$$

T(x,y) adalah thresholding dari piksel (x,y),µ(x,y) adalah *mean gray* ,s(x,y) adalah standar deviasi, dan k adalah koefisien koreksi . Dengan n adalah banyaknya data dan Xi merupakan nilai data ke-i. Sedangkan untuk standar deviasi dapat dihitung dengan fungsi aritmetik sebagai berikut:

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2}$$

Dimana {x1,x2,…,xn} adalaah nilai data dari sampel dan adalah rata rata nilai dari data. Adapun hasil dari T(x,y) merupakan nilai threshold yang didapat dan kemudian dijadikan acuan untuk logika biner.

2.2.2. Deteksi Ukuran Blok

Beberapa tahapan untuk melakukan deteksi ukuran blok adalah mencari turunan pertama dari kurva Hs(y), mencari titik ekstrim dari turunan pertama, mencari nilai dari *inflection point*, dan mencari ukuran blok. Langkah awal yang dilakukan adalah mencari turunan pertama dari kurva Hs (y) . Pencarian turunan pertama dilakukan untuk mendapatkan gradient atau kemiringan kurva. Hal ini dapat menunjukkan karakteristik dari citra sehingga pada proses berikutnya dapat dilakukan pemisahan antara karakter dan *background* berdasarkan intensitas keabuan citra masukan, baik sebagai wilayah karakter atau *background*. Turunan pertama dari kurva Hs(y) dirumuskan oleh persamaan berikut.

$$H_s(y) = \frac{dH_s(y)}{dy} = \frac{H_s(y+1) - H_s(y)}{(y+1) - y} = H_s(y+1) - H_s(y)$$

Langkah berikutnya adalah mencari titik ekstrim dari turunan pertama, yaitu titik dimana kurva asal berada pada nilai maksimum atau minimum. Titik ekstrim ditunjukkan sebagai deret $Y = \{Y_j \mid 0 < j < n\}$ yang dapat didefinisikan sebagai titik dimana nilai turunan pertamanya adalah nol. Setelah didapatkan deret Y sebagai titik ekstrim, maka langkah selanjutnya yaitu mencari *inflection point*. *Inflection point* { } $P = \{P_j \mid 1 < j < n\}$ didapatkan antara indeks Y_j *dan* Y_j+1. Berikut ini adalah perumusan *inflection point* yang digunakan untuk proses deteksi ukuran blok.

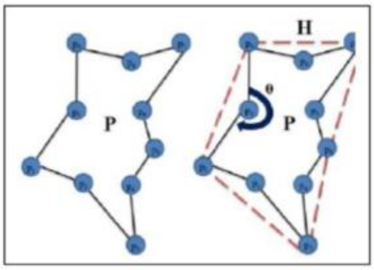
$$P_j = Max_{Y_j < Y < Y_{j+1}}\{|H_s\ |\}$$

Dimana H'_s merupakan nilai turunan pertama yang telah didapatkan pada kurva sebelumnya. Nilai turunan yang maksimal pada kurva turunan menunjukkan bahwa *gradient*/kemiringan paling besar.

2.3. *Convex Hull*

Convex hull merupakan persoalan klasik dalam komputasi geometri. *Convex hull* digambarkan secara sederhana dalam sebuah bidang sebagai pencarian subset dari himpunan titik pada bidang tersebut, sehingga jika titik-titik tersebut dijadikan poligon maka akan membentuk poligon yang konveks . Suatu poligon dikatakan konveks jika garis yang menghubungkan antar kedua titik dalam poligon tersebut tidak memotong garis batas dari poligon. *Convex hull* suatu obyek P didefinisikan sebagai area poligon convex terkecil yang melingkupi P. Oleh karena itu, untuk suatu himpunan titik N {p0, p1, p2, . . . , pN}ϵ P, maka dapat dinyatakan bahwa *hull* H dapat

disusun dengan M titik dari himpunan N untuk membuat suatu area konveks poligon minimum. Pada dasarnya logika *convex hull* terlihat seperti gambar 2 berikut:



Gambar 2.1 Generalisasi Logika Convex Hull

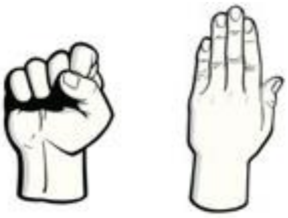
Convex hull pada gambar 2.1 dapat dibuat dengan mengambil sudut interior θ, dari tiga titik yang bersebelahan {p1, p0, p9}. Jika θ > π maka p0 dianggap sebagai titik *reflex* dan p0 bukan anggota M. Himpunan akhir H adalah {p1, p9, p7, p5, p3} . Pencarian convex hull dari sebuah himpunan titik Q (CH(Q)) yaitu dengan mencari sebuah *convex* set terkecil yang memuat seluruh titik pada Q. *Convex hull* dari sebuah himpunan titik Q (CH(Q)) pada n dimensi adalah seluruh irisan dari semua *convex* set yang mengandung Q. Sehingga untuk N buah titik {p1, p2, ..., pN}€ P, *convex hull* merupakan himpunan *convex* combination yang dinyatakan dengan,

$$CH\ (Q) = \left\{ \sum_{j=1}^{N} \lambda_j\ p_j;\ \lambda_j \geq 0;\ \sum_{j=1}^{N} \lambda_j\ = 1 \right\}$$

Convexity defect: adalah sebuah fitur dalam *OpenCV* yang berfungsi untuk menemukan *defect* antara *convex hull* yang terbentuk dengan kontur dari poligon. *Defect* tersebut berguna untuk menemukan *feature* pada sebuah poligon, salah satunya yaitu untuk mendeteksi jari tangan manusia.

2.4. ***Hand Gesture***

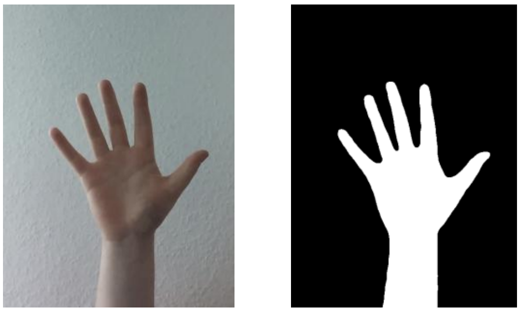
Gestur tangan adalah bagian yang sangat penting dari gerakan bahasa tubuh. Gerakan ini paling membantu ketika seseorang berbicara dengan orang yang berbahasa lain. Arti dari gestur tangan dalam suatu budaya yang berkembang pada suatu tempat bisa jadi diterjemahkan berbeda pada budaya yang lain. Adapun contoh *hand gesture* seperti gambar 2.3 berikut ini.



Gambar 2.3 Contoh *Hand gesture*

2.4.1. Segmentasi Citra Isyarat Tangan

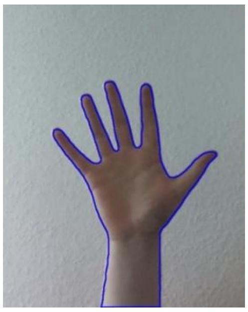
Segmentasi yang bertujuan untuk memisahkan objek pada citra dengan background berikut ini untuk memperoleh objek berusa isyarat tangan. Pemrosesan dengan berdasarkan segmentasi warna dimana memisahkan pola warna kulit dengan background. Adapun contoh segmentasi objek isyarat tangan pada citra digital adalah seperti pada gambar 2.4 berikut.



Gambar 2.4 Segmentasi objek isyarat tangan pada citra digital.

2.4.2. **Mendeteksi Kontur Isyarat Tangan**

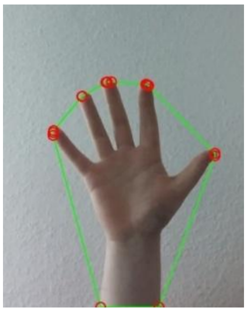
Pemrosesan dalam mendeteksi kontur untuk mendeteksi pola objek pada tangan yang ditunjukkan. Adapun contoh mendeteksi kontur obyek seperti pada gambar 2.5 berikut.



Gambar 2.5 Kontur objek isyarat tangan.

2.4.3. Pendeteksian Ujung Jari

Pendeteksian ujung jari ditandai dengan menampilkan informasi berupa tanda pada citra isyarat tangan pada ujung jari. Tanda tersebut membentuk ,poligon sebagai pola ujung jari.



Gambar 2.6 Pendeteksian ujung jari. .

2.5. Pengenalan Pola

Pengenalan pola merupakan bidang dalam pembelajaran mesin dan dapat diartikan sebagai "tindakan mengambil data mentah dan bertindak berdasarkan klasifikasi data" Dengan demikian pengenalan pola merupakan himpunan kaidah bagi pembelajaran diselia (*supervised learning*). Ada beberapa definisi lain tentang pengenalan pola, di antaranya:

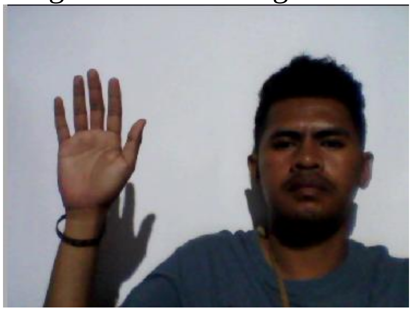
1. Penentuan suatu objek fisik atau kejadian ke dalam salah satu atau beberapa kategori.

2. Ilmu pengetahuan yang menitikberatkan pada deskripsi dan klasifikasi (pengenalan) dari suatu pengukuran.
3. Suatu pengenalan secara otomatis suatu bentuk, sifat, keadaan, kondisi, susunan tanpa keikutsertaan manusia secara aktif dalam proses pemutusan.

Berdasar beberapa definisi di atas, pengenalan pola bisa didefinisikan sebagai cabang kecerdasan yang menitik-beratkan pada metode pengklasifikasian objek ke dalam klas - klas tertentu untuk menyelesaikan masalah tertentu.

2.5.1. Training Pola isyarat tangan.

Pertama kali dalam sistem pengenalan pola adalah menentukan pola angka dalam isyarat tangan. Pola tersebut digunakan untuk menentukan inputan selanjutnya dalam memproses pola obyek menjadi informasi pola angka isyarat tangan. Adapun pola isyarat tangan dalam training adalah sebagai berikut.



Terbuka

Tertutup

Gambar 2.7 Trainning Pola

2.6. Perancangan Sistem Aplikasi

Variabel yang menyusun program aplikasi yang digunakan sebagai perangkat dalam penelitian.

a. Segmentasi obyek tangan

   Segmentasi pada obyek ditangkap dari capture webcam yang bekerja secara streaming. Proses segmentasi terdiri dari memisahkan warna obyek, kemudian segmentasi obyek bagian tangan. Hasil pre-processing ini berupa kelompok obyek tangan.

b. Deteksi kontur obyek tangan

   Obyek telapak tangan selanjutnya diproses dengan menelusuri kontur (tepi obyek) melalui perbedaan warna obyek dan background. Hasil pemrosesan pada deteksi kontur tangan berupa data nemerik koordinat piksel kontur.

c. Analisis Citra

   Pose tangan berikut bekerja dengan memproses kontur obyek tangan. Proses ini menghitung jumlah titik tertinggi pada kontur yang merupakan fitur jari sebagai kondisi isyarat tangan yang terbuka memiliki lebih dari satu titik tinggi pada dan tertutup hanya memiliki satu titik tinggi pada kontur.

Berikut ini adalah penjabaran perancangan sistem pengenalan hand gesture dengan input dinamis pada webcam dalam bentuk *flowchart*. Perancangan sistem berikut ini terdiri dari pemrosesan dengan tahap-tahap sebagai berikut.

a. Input

Input program aplikasi berikut ini merupaakan capture pada sinyal digital yang ditampilkan oleh optik webcam pada program aplikasi. Selanjutnya program aplikasi melakukan scanning piksel untuk memproleh data numerik intensitas piksel.

b. *Thresholding* (algoritma *Ni-Black*)

Thresholding berikut ini untuk memisahkan background dan obyek dengan nilai ambang yang diperoleh dari algoritma *Ni-Black*. Proses ini menghasilkan segmen obyek dengan warna hitam sebagai obyek dan background dengan warna putih.

c. Eliminasi *Noise*

Proses eliminasi *noise* adalah menghapus titik piksel obyek yang dianggap sebagai *noise*. Proses ini menghitung konektivitas piksel dalam spatial filter 3x3.

d. Segmentasi obyek telapak tangan

Proses segmentasi obyek berikut ini memisahkan obyek telapak tangan dengan obyek lainnya yang bukan termasuk telapak tangan. hasil pemrosesan pada tahap ini adalah segmen ditampilkan dengan menandai area telapak tangan.

e. Deteksi kontur telapak tangan

Proses deteksi kontur berikut ini menelusuri tepi obyek telapak tangan. data kontur telapak tangan tersebut berupa koordinat piksel tepi obyek telapak tangan.

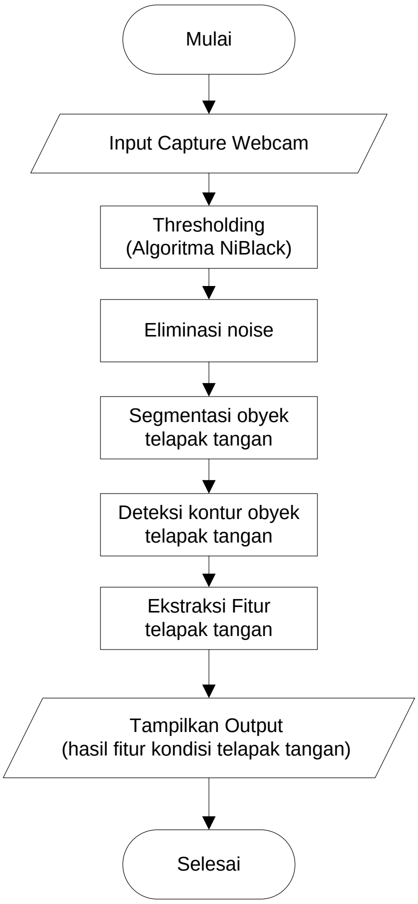
f. Ekstraksi fitur isyarat tangan

Proses berikut ini menghitung titik tinggi pada koordinat kontur dimana jumlah titik-titik tersebut merupakan fitur pada jari yang menentukan kondisi isyarat tangan terbuka atau tertutup.

g. Tampilkan *Output*

Output program aplikasi adalah teks yang berupa kondisi isyarat tangan terbuka dan tertutup yang merupakan hasil pemrosesan secara streaming.

Adapun penjabaran *flowchart* dalam perancangan sistem aplikasi dalam pengenalan isyarat tangan dengan input *hand gesture* dinamis seperti pada gambar 3.1 berikut.



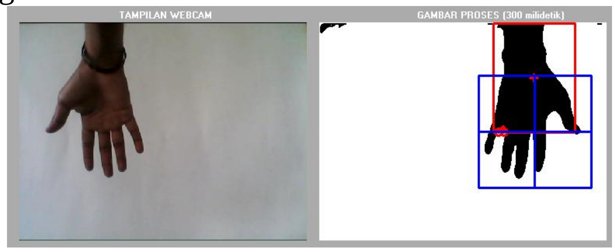
Gambar 3.1 *Flowchart* pemrosesan sistem pengenalan isyarat tangan

3. HASIL DAN PEMBAHASAN

4.1.1. Segmentasi Tangan *Real-Time*

Pengujian berikut pada segmentasi obyek dengan ragam pose tangan. pengujian ini untuk mengetahui efektifitas program aplikasi dalam memperoleh segmen tangan.

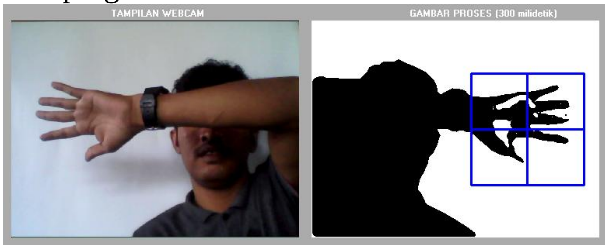
1. Pose frontal tangan terbalik



Gambar 4.1 Pengujian dengan pose tangan terbalik

Penjelasan : Program aplikasi tidak dapat mensegmen pada obyek tangan dari *lock* pose sehingga hasil pendeteksian tidak efektif.

2. Pose tangan dari samping

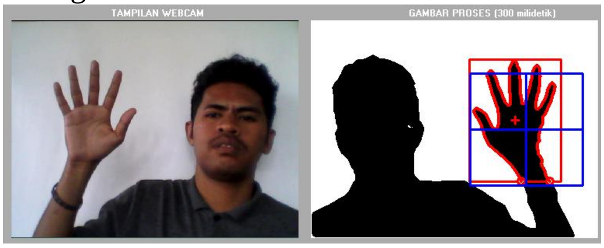


Gambar 4.2 Pengujian dengan pose tangan dari samping kiri

Gambar 4.3 Pengujian dengan pose tangan dari samping kanan

Penjelasan : Program aplikasi tidak dapat mensegmen pada obyek tangan dari *lock* pose samping kiri dan samping kanan sehingga hasil pendeteksian isyarat tangan tidak efektif.

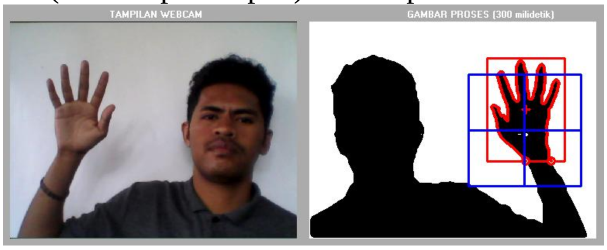
3. Pose tangan *forntal* tegak lurus



Gambar 4.4 Pengujian dengan pose tangan tegak lurus

Penjelasan : Posisi forntal tangan pada webcam adalah posisi terbuka secara berhadap-hadapan dengan kamera. Program aplikasi dapat memproleh segmen pada obyek tangan dari lock dengan pose tegak lurus dan berhasil mengenali isyarat tangan.

4. Pose tangan *frontal* (berhadap-hadapan) terhadap kamera



Gambar 4.5 Pengujian dengan pose frontal terhadap kamera

Penjelasan : Program aplikasi dapat memproleh segmen pada obyek tangan dari *lock* dengan pose frontal menghadap kamera dan berhasil mengenali isyarat tangan.

5. Pose tangan *frontal* dengan obyek dua tangan

Gambar 4.6 Pengujian dengan obyek dua tangan

Penjelasan : Program aplikasi tidak dapat memproleh segmen pada obyek tangan dengan obyek dua tangan sehingga tidak efektif mengenali isyarat tangan.

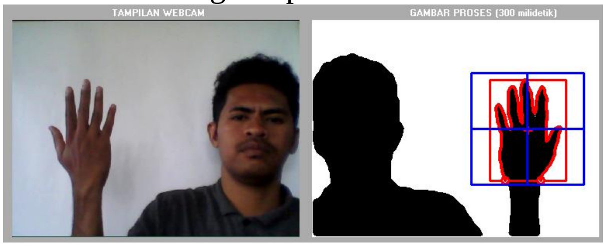
6. Pose tangan tidak *frontal* menghadap kamera

Gambar 4.7 Pengujian dengan pose tangan miring

Gambar 4.8 Pengujian dengan pose tangan agak miring

Penjelasan : Program aplikasi tidak dapat memproleh segmen pada obyek tangan yang terbuka tetapi miring terhadap kamera sehingga tidak efektif mengenali isyarat tangan.

7. Pose tangan *frontal* terbalik menghadap kamera



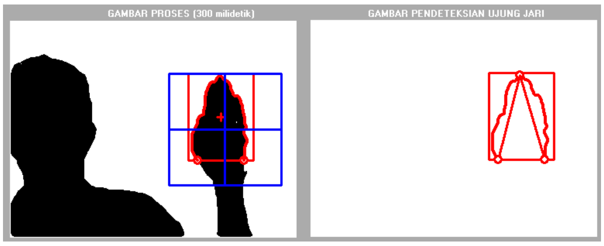
Gambar 4.9 Pengujian dengan pose tangan terbalik

Penjelasan : Program aplikasi dapat memproleh segmen pada obyek tangan dari *lock* dengan pose *frontal* menghadap terbalik pada kamera dan berhasil mengenali isyarat tangan.

4.1.2. Pengenalan isyarat tangan.

Pengujian berikut pada pengenalan isyarat tangan dengan program aplikasi yang sudah memperoleh segmen obyek tangan dengan pose frontal meghadap kamera.

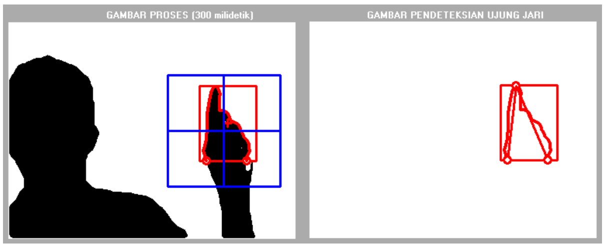
1. Pengujian dengan pose terbuka dengan jari yang dirapatkan



Gambar 4.10 Pengujian dengan pose tangan terbuka rapat

Penjelasan : Program aplikasi dapat dapat mendeteksi ujung jari tetapi tidak dapat menghitug jari yang tertutup dalam isyarat tangan.

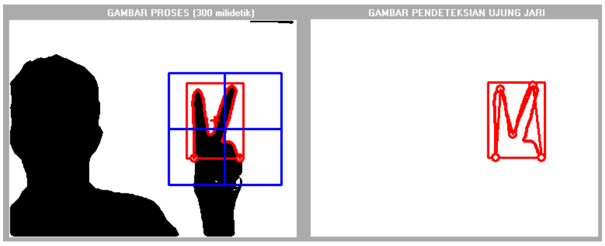
2. Pengujian dengan pose terbuka dengan jari yang terbuka
   a. Tangan terbuka dengan satu jari



Gambar 4.11 Pengujian dengan pose tangan terbuka satu jari

Penjelasan : Program aplikasi dapat dapat mendeteksi ujung jari dan menghitug satu jari yang terbuka dalam isyarat tangan.

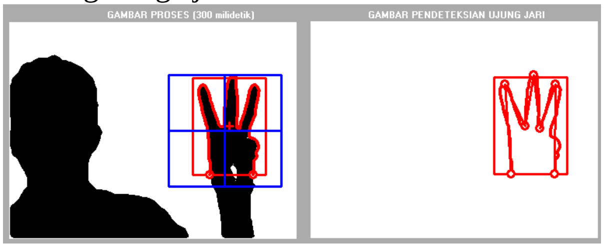
   b. Tangan terbuka dengan dua jari



Gambar 4.12 Pengujian dengan pose tangan terbuka dua jari

Penjelasan : Program aplikasi dapat dapat mendeteksi ujung jari dan menghitug dua jari yang terbuka dalam isyarat tangan.

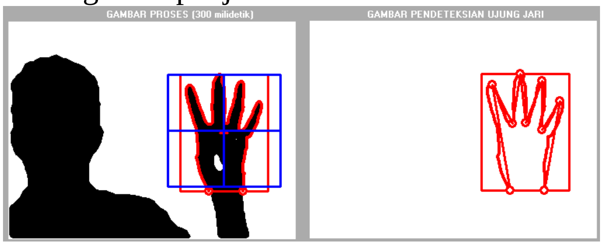
   c. Tangan terbuka dengan tiga jari



Gambar 4.13 Pengujian dengan pose tangan terbuka tiga jari

Penjelasan : Program aplikasi dapat dapat mendeteksi ujung jari dan menghitug tiga jari yang terbuka dalam isyarat tangan.

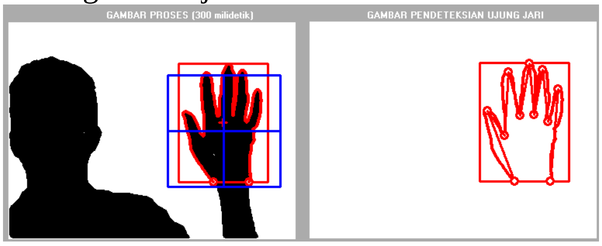
d. Tangan terbuka dengan empat jari



Gambar 4.14 Pengujian dengan pose tangan terbuka empat jari

Penjelasan : Program aplikasi dapat dapat mendeteksi ujung jari dan menghitug empat jari yang terbuka dalam isyarat tangan.

e. Tangan terbuka dengan lima jari

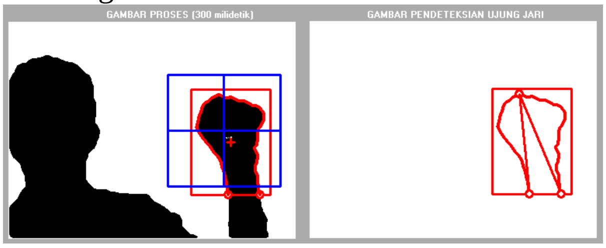


Gambar 4.15 Pengujian dengan pose tangan terbuka lima jari

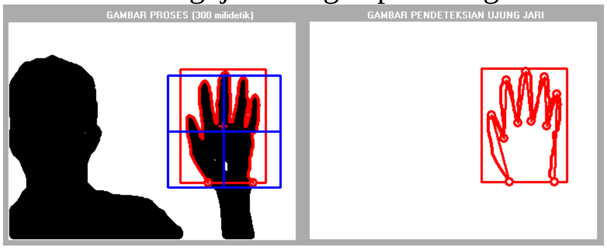
Penjelasan : Program aplikasi dapat dapat mendeteksi ujung jari dan menghitug lima jari yang terbuka dalam isyarat tangan.

4.1.3. Pengenalan Isyarat Tangan Terbuka dan Tertutup

Berikut adalah pengujian pada pose tangan yang tertutup dan terbuka. Adapun hassil pengujian adalah sebagai berikut.



Gambar 4.16 Pengujian dengan pose tangan tertutup



Gambar 4.17 Pengujian dengan pose tangan terbuka

Penjelasan : Program aplikasi dapat dapat segmentasi obyek tangan untuk mendeteksi ujung jari dan berhasil mengenali isyarat tangan terbuka dan tertutup.

4.1.4. Kebutuhan Waktu Pemrosesan.

Pengujian berikut pada komputer dengan prosessor i5 (2,50 Ghz), ram 4 Gb dan Vga 1024 Mb. Pengujian dengan menuliskan waktu setiap pemrosesan pada *frame* hasil

capture kemudian dirata-rata untuk memperoleh kebutuhan waktu selama pemrosesan. Adapun hasil pengujian dengan rata-rata pengujian adalah sebagai berikut.

Tabel 4.1 Kebutuhan waktu pemrosesan

No	Frame Pemrosesan	Waktu Pemrosesan (milidetik)
1	frame 1	55
2	frame 2	67
3	frame 3	62
4	frame 4	56
5	frame 5	46
6	frame 6	60
7	frame 7	47
8	frame 8	52
9	frame 9	54
10	frame 10	52
rata-rata		55,1

Penjelasan : Diperoleh rata-rata hasil pemrosesan adalah 55.1 milidetik untuk setiap pemrosesan pada citra hasil capture.

4. KESIMPULAN DAN SARAN

4.1. Kesimpulan

Adapun kesimpulan dalam tugas akhir berikut ini adalah sebagai berikut.

1. Pose tangan frontal dan berhadap-hadapan pada lock area tidak tepat menghasilkan segmentasi obyek isyarat tangan dengan pose tangan terbalik, dari samping kiri dan kanan, dengan dua tangan dan pose tangan tidak frontal menghadap miring pada kamera.
2. Sedangkan pose tangan teoat menghasilkan segmen obyek isyarat tangan dengan pose frontal (berhadap-hadapan) tegak lurus dan terbalik menghadap kamera.
3. Pengenalan isyarat tangan dapat mengenali pose jari terbuka dan tidak mampu menghitung ujung jari yang terbuka dalam pose jari yang rapat.
4. Pengenalan isyarat tangan efektif dengan pose jari terbuka dengan isyarat jari dua, tiga, empat dan lima jari sedangkan pose satu jari dikenali dalam isyarat tangan tertutup.
5. Pengenalan isyarat tangan yang tepat dalam segmentassi tangan dengan jari tidak rapat telah berhasil mengenali pose tangan terbuka dan tertutup.

4.2.Saran

Program aplikasi pengenalan isyarat tangan dengan input dinamis masih mnegalami kendala untuk segmentasi dengan beberapa pose tangan yang tidak tegak lurus dan frontal menghadap kamera, sehingga perlu dikembangkan dalam proses segmentasi yang efektif untuk memperoleh obyek isyarat tangan yang mampu menopang pengenalan isyarat tangan. Proses mengenali isyarat tangan dengan

menghitung ujung jari yang terbuka masih mengalami kendala dalam pose jari yang rapat. Pengembangan penelitian juga diperlukan untuk mendeteksi jari yang terbuka dengan pose yang rapat.

.

5. DAFTAR PUSTAKA

[1] Arsyah , Ulya Ilhami. 2018. *Penerapan Aplikasi Multimedia Mengenal Bahasa Isyarat Sebagai Pendukung Pembelajaran Bagi Tuna Rungu*. (Jurnal Manajemen Informatika dan Teknik Komputer. Volume 3, Nomor 2.

[2] Asmara, Rosa Andrie, ST., MT., Dr. Eng. 2010. *Pengolahan Citra Digital*. Polinema Press : Malang.

[3] Baroto, Adi Kusumo. 2018. *Bahasa Isyarat SIBI dan Bisindo, Tilik Perbedaannya.* (Tempo.co : 2018)

[4] Darmawan, M. Wirman. 2009. *Identifikasi Mutu Buah Mangga Arum Manis Berdasarkan Warna Menggunakan Image Processing dan JST.* (Yogyakarta: Jurusan Teknik Elektro Universitas Gadjah Mada)

[5] Putra, Darma. 2010. *Pengolahan Citra Digital.* (Andi Offset : Yogyakarta).

[6] Rasul, Juharis, Abdul Hamid. 2007. *Teknologi Informasi dan Komunikasi*. (Yudhistira : Bogor).

[7] Utami , Dina Budhi, Muhammad Ichwan. 2017. *Pengenalan Pose Tangan Menggunakan HuMoment* . (Jurnal Infotel : Bandung)

[8] Widyarini , Sekaring Tyas. 2015. *Pemrograman Matlab Untuk Pengolahan Citra Digital*. (UB press : Malang).