



GERAKAN ANIMASI MANUSIA BERDASARKAN ISYARAT TANGAN BERBASIS PENGOLAHAN CITRA

Gustina¹, Ahmad Husna Ahadi², Ridwan³, Yandi Anzari⁴, Fattachul Huda Aminuddin⁵

¹Program Studi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Nurdin Hamzah

²Program Studi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Nurdin Hamzah

³Program Studi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Nurdin Hamzah

⁴Program Studi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Nurdin Hamzah

⁵Program Studi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Nurdin Hamzah

E-mail: gus17tina08@gmail.com

Article History:

Received: 06-06-2024

Revised :26-06-2024

Accepted:09-07-2024

Keywords: deteksi tangan; HuMoments; animasi 3D

Abstract: Berkembangnya teknologi komputer grafik berpengaruh pada peningkatan pembuatan obyek animasi. Kendali gerakan dari obyek animasi tersebut juga beragam dan berkembang diantaranya menggunakan teknologi pengolahan citra dari gambar tangkapan kamera. Namun penggunaan kendali obyek animasi dengan gerakan tangan masih jarang sekali digunakan. Makalah ini dibuat dengan tujuan memberikan solusi alternatif untuk mengatasi permasalahan tersebut, yaitu membangun sebuah software aplikasi kendali gerakan animasi manusia 3D berdasarkan isyarat tangan yang telah di tangkap oleh kamera (webcam). Obyek isyarat tangan yang ditangkap kamera dikodekan, hasil pengkodean disesuaikan dengan data pada database yang berisi simbol kode hasil HuMoments berserta arti terjemahannya. Ada empat macam simbol kode yang ada pada database yaitu isyarat tangan dalam bentuk jari tangan Gun, Thumb, Phone dan Devil. Diperoleh hasil pengujian deteksi isyarat tangan yang terbaik pada intensitas cahaya ruang 450 lux di malam hari dengan jarak tangan terhadap kamera 100 cm kecepatan deteksi 3.61s, bentuk isyarat tangan yang tercepat dikenali pada berbagai kondisi adalah "Gun" dengan waktu rata-rata 59.98s sebagai pembangkit animasi manusia 3D dengan gerakan jalan ke kiri.

PENDAHULUAN

Berkembangnya teknologi komputer grafik berpengaruh pada peningkatan pembuatan obyek animasi. Demikian pula teknologi pengolahan citra dari gambar tangkapan kamera. Seiring dengan meningkatnya teknologi grafik dan meningkatnya hardware untuk mempercepat proses kecepatan visualisasi yang sangat tinggi pula. Pada makalah ini dikembangkan software aplikasi gerakan animasi manusia menggunakan tangan. Software ini menampilkan karakter animasi 3D berupa manusia yang bisa bergerak. Animasi manusia tersebut bisa bergerak dengan berjalan ke kanan, ke kiri, maju dan mundur. Untuk menggerakkan animasi manusia tersebut adalah dengan menggunakan isyarat tangan dalam bentuk jari tangan Gun, Thumb, Phone dan Devil.

Software aplikasi ini sebagai alat bantu untuk menterjemahkan isyarat tangan secara komputerisasi yang akan ditampilkan dalam bentuk teks yaitu abjad "A" untuk isyarat tangan bentuk Thumb, "L" sebagai isyarat tangan Gun, "Y" adalah isyarat tangan Phone dan teks "I Love You" adalah isyarat tangan bentuk Devil yang terdapat dalam database. Sedangkan webcam digunakan sebagai perangkat untuk melakukan pengambilan gambar isyarat tangan, sehingga gambar yang telah tertangkap oleh webcam diproses dengan mengidentifikasi background dari obyek untuk mengambil bagian objek yang dibutuhkan, mengidentifikasi contour dari obyek yang dimaksud, dan menkodekan hasil contour dengan proses HuMoment, sehingga dapat diterjemahkan dalam bentuk teks. Apabila teks yang keluar ada dalam database maka akan di bangkita sebuah obyek animasi manusia 3D dengan gerakan tertentu.

METODE PENELITIAN

A. Deteksi Objek

penelitian ini dirancang untuk menggambarkan langkah-langkah yang sistematis dan terstruktur Pengambilan gambar isyarat tangan dilakukan menggunakan webcam dan hasilnya akan diterjemahkan dalam bentuk abjad tertentu

1. Segmentasi Background

Inisialisasi background dapat dilakukan dengan Averaging Background Method yang pada dasarnya mempelajari tentang rata-rata dan standart deviasi dari setiap piksel sebagai model dari background. Averaging Background Method adalah menghitung perbedaan rata-rata antara setiap piksel gambar dengan model background. Model dari background dibuat dengan satu set frame dan membedakan antara frame terakhir dengan cyAcc dan cyAbsDiff dari OpenCV library. Dimanan langkah-langkahnya sebagai berikut :

- Pengalokasian gambar
- Pengambilan frame-frame pada background
- Penghitungan rata-rata perbedaan nilai frame tiap piksel
- Menentukan nilai batas atas berdasarkan perbedaan nilai rata-rata tiap frame
- Segmentasi gambar foreground (segala sesuatu yang bukan termasuk bagian dari background)

$$Y=0.299 R + 0.587 G + 0.114 B \quad (1-1)$$

$$U=(B-Y) \times 0.493 \quad (1-2)$$

$$V=(R-Y) \times 0.877 \quad (1-3)$$

- Penghapusan memori gambar setiap kali selesai menggunakan background

2. Reprerentasi Warna

- RGB (*Red, Green, Blue*)

Data video dapat dipisahkan menjadi komponen-komponen untuk masing-masing warna, yaitu merah (red), hijau (green) dan biru (blue). Warna tiap piksel ditentukan oleh kombinasi intensitas dari masing-masing komponen warna. Sebagai contoh, pada RGB 24 bit, masing-masing komponen warna dinyatakan dalam 8 bit atau 256 level. Contoh, warna biru langit direpresentasikan dengan R=181, G=189, dan B=249. RGB dikategorikan sebagai 3 channel.

- YUV

Model warna ini dikembangkan untuk mengantisipasi perkembangan informasi berbasis video. Secara umum, dapat dikatakan bahwa model warna ini merupakan bagian dari ruang warna transmisi video dan televisi. Model lain yang mirip dengan YUV adalah YCrCb, perbedaannya terletak bahwa YUV adalah system warna analog sedangkan YCrCb adalah sistem warna digital. YUV dikategorikan 1 channel. Persamaan transformasi warna RGB ke YUV

3. ROI (*Region Of Interest*)

ROI dapat di-set maupun reset dengan opencv. Untuk mengaktifkannya dapat menggunakan `cvSetImageROI()`, untuk me-nonaktifkan dapat menggunakan `cvResetImageROI()`, dan untuk membuat bentuk ROI menjadi persegi, maka dapat digunakan `cvRect()`.

Pada software ini, ROI digunakan untuk menempatkan tangan/jari yang merupakan obyek isyarat tangan dan dipastikan bahwa tidak ada objek foreground selain tangan. Selanjutnya sistem akan dengan mudah mengidentifikasi countour dari tangan tersebut tanpa ada gangguan dari obyek yang lain.

4. *Conture Finding*

Contour sebenarnya adalah sebuah nilai sequences, maka segala sesuatu harus berada pada koordinat sequencesi. Contour ditemukan jika terdapat perbedaan titik-titik yang tinggi dengan tetangganya. Hal ini terjadi karena didasarkan pada hasil deteksi tepi. Untuk mendapatkan hasil contour melalui OpenCV, dapat dilakukan dengan menuliskan fungsi `cvFindCountour()` dan `cvDrawContour()`.

5. *HuMoments*

Salah satu cara untuk untuk membandingkan dua contour adalah dengan menghitung nilai contour moment. Perhitungan contour dilakukan dengan mengintegrasikan atau menjumlahkan contour setiap piksel. Secara umum didefinisikan bahwa (p,q) contour seperti persamaannya

$$m_{p,q} = \sum_{x,y} I(x,y) x^p y^q$$

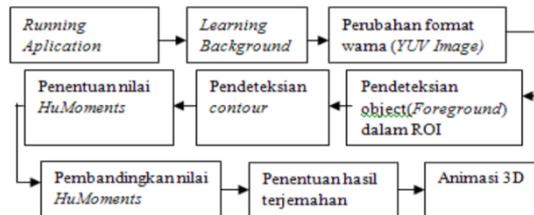
B. Perancangan Sistem

Tahap paling awal yang dilakukan pada penelitian ini adalah persiapan peralatan-peralatan yang dibutuhkan dalam sistem, yaitu sebuah komputer atau laptop dengan kamera. Adapun spesifikasinya seperti pada tabel 1.

TABEL I. SPESIFIKASI PERALATAN

Deskripsi	Spesifikasi Minimum
Processor	AMD Turion™ X2 Dual-Core Mobile RM-75,MMX, 3DNow
Memory	766MB RAM
DirectX Version	DirectX 9.0c
Operating System	Microsoft Windows XP SP2
Webcam	Fitur dari laptop(webcam Crystal Eye)

Proses deteksi obyek tangan sampai pembangkitan obyek animasi 3D diperlihatkan pada gambar 2.



Gambar 2. Blok diagram sistem

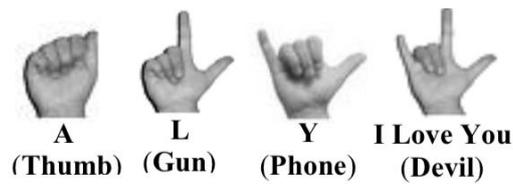
Pada Gambar 2, sebelum melakukan pendeteksian obyek dilakukan terlebih dahulu proses capturing object selanjutnya dilakukan learning background untuk mempermudah pencarian object tangan/jari (Foreground). Untuk pendeteksian background dilakukan dengan metode Averaging Background. Pendeteksian obyek hanya dibatasi oleh ROI, sehingga untuk pemrosesan selanjutnya hanya berada pada area ROI. Hal ini untuk mempermudah pengambilan obyek yang diinginkan. Setelah itu, dilakukan pencarian contour dari foreground. Dari hasil contour tersebut yang memiliki format warna RGB, akan diubah ke dalam format warna YUV, sehingga hasil keluarannya akan berwarna untuk foreground adalah putih dan background adalah hitam. Kemudian dilakukan pencarian nilai HuMoments dari countour yang telah didapatkan sebelumnya. Nilai HuMoments dari hasil capture akan dibandingkan dengan nilai HuMoments yang telah terinisialisasi pada program yang telah dilakukan sebelumnya. Dimana hampir semua nilainya berada pada kisaran nilai antara 0 sampai 1. Jika, nilainya sama, maka hasil terjemahan dalam bentuk animasi dapat ditampilkan.

1. Objek Isyarat Tangan

Dalam perencanaan pembuatan penelitian ini, objek yang akan digunakan adalah tangan (telapak tangan dan jari) yang digunakan sebagai perantara untuk isyarat tangan dengan huruf (alphabet).

Fungsi isyarat tangan bentuk thumb untuk gerakan jalan ke belakang/mundur, bentuk gun untuk gerakan jalan ke kiri, bentuk phone untuk gerakan ke depan/maju dan bentuk devil untuk gerakan ke kanan.

2. Animasi 3D



Gambar 3. Bentuk isyarat tangan

Pembuatan model objek 3 dimensi dalam dunia virtual menggunakan pemrograman VRML (Virtual Reality Modeling Language). Animasi yang telah dibuat dengan gerakan berjalan ke kiri, ke kanan, maju dan mundur. Untuk memperoleh tampilan tersebut harus diatur translasi, rotasi dan viewpointnya.

3. Pembentukan Rectangle (ROI) pada Window

Pembentukan rectangle ini akan selalu ter-display pada aplikasi ini selama camera berada pada posisi ON. Pembuatan rectangle pada window aplikasi mempermudah user meletakkan obyek tangan/jari. Besar luasan dari rectangle ini sama dengan luasan ROI yang akan dibuat.



Gambar 4

Titik awal koordinat sumbu "x" adalah 20 dari tepi samping frame, sedangkan titik awal koordinat sumbu "y" adalah 100 dari tepi atas frame. Untuk panjang dan lebar ROI, masing-masing bernilai 250. Pembuatan sebuah rectangle dilakukan dengan menggabungkan empat koordinasi untuk keempat tepi ROI, yaitu "p1.x ; p1.y ; p2.x ; p2.y".

4. Proses Learning Background

Proses learning background dilakukan dengan beberapa step. Untuk fungsi `AllocateImage(rawImage)` merupakan suatu fungsi yang dijadikan sebagai pengalokasian variabel-variabel yang akan digunakan dalam proses learning background. Setelah dialokasikan, maka akan didefinisikan nilai skala tinggi dan skala rendah. Dimana `scaleHigh` telah didefinisikan bernilai 7.0 sedangkan `scaleLow` telah didefinisikan bernilai 6.0. Setelah semuanya telah terdefiniskan, maka baru dilakukan learning background dengan melakukan `accumulateBackground(rawImage)`. Pertama dilakukan perubahan untuk raw background kedalam bentuk float. Kemudian semua nilai floating-point pada gambar yang tersimpan dalam variabel `Iscratch` dikumpulkan (`cvAdd`) ke variabel `Iavg`. Kemudian dilakukan perhitungan nilai tiap-tiap frame dengan menggunakan `cvAbsDiff` dan mengumpulkannya kedalam variabel gambar `IdiffF`.

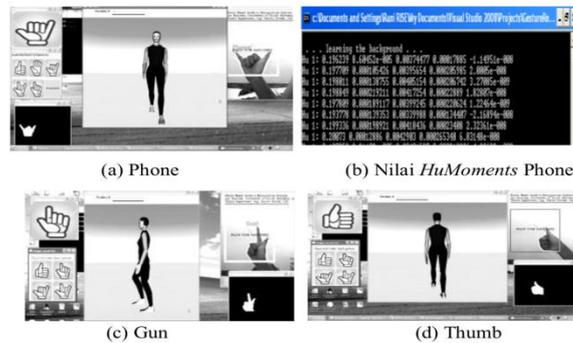
5. Proses Perubahan Format RGB menjadi YUV

Perubahan format RGB menjadi YUV dengan menggunakan salah satu sintak OpenCV, yaitu "`cvCvtColor(rawImage,yuvImage,CV_BRG2YCrCb.`". Maksud dari

sebaris sintak tersebut yaitu dilakukannya perubahan format dari variable gambar rawImage yang berupa RGB dan menggubahnya menjadi YUV dan hasilnya disimpan kedalam variable yuvImage.

6. Proses Pendeteksian Obyek (Foreground) dalam ROI

Setelah melakukan konversi format warna, maka proses akan dilanjutkan menuju fungsi auxfindforegroun. Dimana fungsi ini akan berisi program atau syntax mengenai



pengenalan foreground atau obyek yang bukan bagian dari inialisasi background pada proses sebelumnya.

7. Proses Pendeteksian Contour

Proses pengambilan contour dengan salah satu perintah dari open CV, yaitu cv Find Contour. Sedangkan untuk melukiskan contour pada monitor aplikasi dilakukan dengan menggunakan perintah cvDrawContour yang didalamnya terdapat CV_RGB(0,255,0) dan nilai inilah yang menjadikan contour terlihat berwarna hijau.

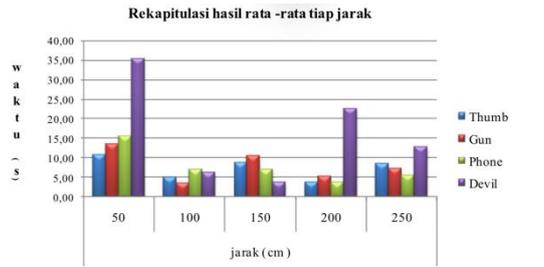
8. Proses Penentuan dan Perbandingan Nilai HuMoments

Penentuan nilai HuMoments ini dilakukan sebagai nilai pembanding untuk nilai HuMoments berikutnya, atau dapat dikatakan bahwa nilai ini merupakan nilai kode database dari masing-masing bentuk tangan. Nilai HuMoments tersebut diambil dari contour yang telah didapatkan. CvHuMoments memiliki parameter nilai yang diambil dari nilai moments dari CvMoment. Nilai HuMoments yang dideklarasikan sebagai database hanya meliputi L1,H1,L2,H2,L3,dan H3. Hal ini dimaksudkan untuk memperkecil adanya nilai yang sama didalam rentang nilai yang berbeda antara gesture tangan yang satu dengan tangan yang lainnya.

HASIL DAN PEMBAHASAN

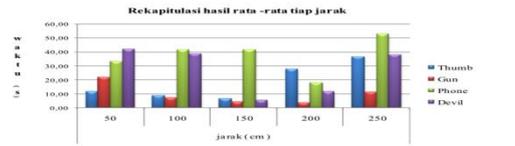
Pengujian dilakukan terhadap perbedaan jarak dan intensitas cahaya, perbedaan background serta perbedaan bentuk tangan manusia, alat ukur intensitas menggunakan Lux meter. Tiap pengujian dilakukan dalam ruangan sebanyak tiga kali pengambilan data.

Pengujian keseluruhan sistem pada beberapa kondisi adalah sebagai berikut:
Hasil pengujian pada malam hari.



Gambar 7. Grafik deteksi tangan dengan intensitas cahaya 450 lux

Hasil pengujian pada siang hari



Gambar 8. Grafik deteksi tangan dengan intensitas cahaya 280 lux

Hasil pengujian (jarak 100cm) pada perbedaan background dalam ruangan dengan intensitas cahaya 255 lux



Gambar 9. Grafik deteksi tangan pada perbedaan background

Hasil pengujian terhadap perbedaan tiga tangan manusia pada jarak 100cm dalam ruangan dengan intensitas cahaya 255 lx



Gambar 10. Grafik deteksi perbedaan bentuk tangan tiga manusia

Hasil rekapitulasi respon kecepatan dari semua pengujian ditunjukkan pada tabel 2 dan gambar 11.

Hasil rekapitulasi respon kecepatan dari semua pengujian ditunjukkan pada tabel 2 dan gambar 11

TABEL II. RESPON KECEPATAN DETEKSI BENTUK TANGAN

Waktu	Respon Kecepatan Deteksi Bentuk Tangan (s)			
	Thumb	Gun	Phone	Devil
Rata-Rata	69.19	59.98	97.08	1:45.16



Gambar 11. Respon kecepatan deteksi obyek

Berdasarkan tabel 2 dan gambar 11, dari semua pengujian yang telah dilakukan antara lain pengujian terhadap jarak dan intensitas cahaya, pengujian terhadap perbedaan background serta pengujian terhadap perbedaan bentuk tangan manusia diperoleh hasil bahwa bentuk tangan “gun” lebih cepat terdeteksi dengan rata-rata waktu 59.98(s) sedangkan yang paling lama terdeteksi adalah bentuk tangan “Devil” dengan rata-rata waktu 1:45.16(s).

KESIMPULAN

Berdasarkan data hasil pengujian dan analisa yang telah dilakukan dapat diperoleh kesimpulan sebagai berikut :

1. Deteksi isyarat tangan yang terbaik pada intensitas cahaya ruang 450 lux di malam hari, jarak tangan terhadap kamera 100 cm dan kecepatan deteksi 3.61s.
2. Warna background yang terbaik adalah putih, pada background ini bentuk “Gun” dikenali tercepat 5.48s.
3. Bentuk isyarat tangan yang tercepat dikenali pada berbagai kondisi adalah “Gun” dengan waktu rata-rata 59.98s.
4. Bentuk isyarat tangan yang paling lama dikenali pada berbagai kondisi adalah “Devil” dengan rata-rata waktu 1:45.16s.

DAFTAR REFERENSI

- [1] Bradski, Gary; Kahler, Adrian “Learning OpenCV, Computer Vision with the OpenCV Library”, 2008, First Edition, O’Reilly Media, Inc., ISBN: 978-0-596-51613-0
- [2] Canpy, “Pengolahan Citra-Akuisisi Data Citra Video”, 2010
- [3] Gady Agam, “Introduction to Programming with OpenCV”, Departement of Computer Science, 2006
- [4] Yandi Anzari, Yonggi Puriza “Aplikasi dan Game Edukasi Sandi Semaphore Berbasis Multimedia”, EPSILON: Journal of Electrical Engineering and Information Technology 19
- [5] Open Computer Vision library [Http://sourceforge.net/project/opencv/library](http://sourceforge.net/project/opencv/library)
- [6] Noah Kuntz, “Learning OpenCV: Computer Vision with the OpenCV Library-OpenCV Tutorial 8 ”, 2009
- [7] Wirasasti Susanto, “Bahasa Isyarat”, 2007.
- [8] Mgr. Jan Kapoun, “Static Hand Gesture Recognition Software”, University of South Bohemia, České Budějovice, 2010.