

Pembuatan *Software* Aplikasi Gerakan Animasi Manusia Menggunakan Tangan

Flora Pandu Agusta, Miftahul Huda, Akuwan Shaleh
Program Studi Teknologi Telekomunikasi - Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember (ITS) Surabaya
Kampus PENS-ITS, Keputih, Sukolilo, Surabaya.
Telp : +62+031+5947280; Fax. +62+031+5946011
Email : flora.eepis@gmail.com

Abstrak – Perkembangan teknologi citra sekarang ini sangat pesat dan canggih. Seiring dengan meningkatnya teknologi grafik dan meningkatnya *hardware* untuk mempercepat proses kecepatan visualisasi yang sangat tinggi. Apalagi perkembangan teknologi citra di luar negeri sudah sangat pesat sekali. Sedangkan di Indonesia teknologi citra masih kurang berkembang pesat. Permasalahannya teknologi citra di Indonesia belum begitu banyak yang mengembangkannya. Tetapi sebenarnya hasil dari teknologi citra banyak yang menggemari dan banyak manfaatnya. Untuk sedikit mengatasi permasalahan tersebut kami membuat dan mengembangkan teknologi citra yaitu sebuah *software* aplikasi gerakan animasi manusia menggunakan tangan. Mungkin suatu saat nanti *software* ini banyak yang menyukainya dan mengembangkannya lagi.

Proyek akhir ini bertujuan untuk membuat *software* aplikasi animasi manusia dimana animasi tersebut bisa bergerak dengan menangkap gerakan tangan yang tertangkap oleh *webcam*.

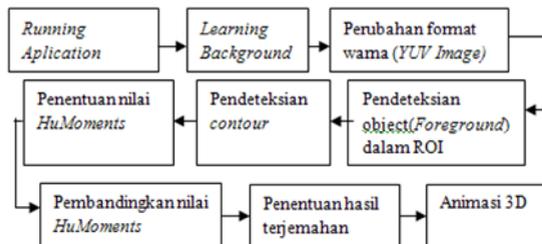
Dari pembuatan Aplikasi ini diharapkan dapat dihasilkan aplikasi yang interaktif. Dari data hasil percobaan tangan dapat terdeteksi dengan cukup baik pada intensitas cahaya 640 lx pada kondisi cahaya ruang pada siang hari dengan jarak 100 cm .

Kata kunci: C++, *OpenCV*, Animasi 3D

A. PENDAHULUAN

Perkembangan teknologi citra sekarang ini sangat pesat dan canggih. Seiring dengan meningkatnya teknologi grafik dan meningkatnya *hardware* untuk mempercepat proses kecepatan visualisasi yang sangat tinggi. Apalagi perkembangan teknologi citra di luar negeri sudah sangat pesat sekali. Sedangkan di Indonesia teknologi citra masih kurang berkembang pesat. Pada proyek akhir ini dikembangkan *software* aplikasi gerakan animasi manusia menggunakan tangan. *Software* ini menampilkan karakter animasi 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 gerakan tangan kita.

B. METODOLOGI



Gambar 2.2. Blok Diagram Sistem

Pada **Gambar 2.2**, sebelum melakukan pendeteksian object melakukan proses *capturing object* maka

dilakukan terlebih dahulu *learning background*. *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. Oleh karena itu, untuk obyek yang merupakan *foreground* hanyalah jari/tangan. Jika ada obyek lain selain jari/tangan dan juga bukan merupakan *background*, maka akan mempengaruhi proses penerjemahan bentuk tangan.

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 *output*-nya akan berwarna untuk *foreground* adalah putih dan *background* adalah hitam. Kemudian dilakukan pencarian nilai *HuMoments* dari *contour* yang telah didapatkan sebelumnya. Setelah itu, 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.

C. PERANCANGAN DAN PEMBUATAN

✓ Bahan dan Alat

Pada sub bab ini akan dijelaskan bahan dan alat apa saja yang akan digunakan dalam pembuatan system proyek akhir ini, diantaranya meliputi :

1. Perencanaan Perangkat Keras
2. Perencanaan Perangkat Lunak

1. Perencanaan Perangkat Keras

Tahap paling awal yang harus dilakukan pada penelitian ini adalah persiapan peralatan-peralatan yang dibutuhkan dalam sistem, yaitu sebuah komputer atau laptop dan kamera internal/eksternal. Pada Proyek Akhir ini digunakan laptop dengan spesifikasi minimum sebagai berikut

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)
Tangan	Telapak tangan dan jari manusia

Tabel Spesifikasi Minimum Peralatan

2. Perencanaan Perangkat Lunak

Perencanaan perangkat lunak yang dibutuhkan dalam sistem ini, antara lain:

a. Microsoft Visual Studio 2008

Sebuah *compiler* yang dibuat menggunakan bahasa pemrograman C++ yang akan digunakan untuk membuat program translasi bahasa isyarat ini.

b. OpenCV

Sebuah *Library* untuk fungsi-fungsi *computer vision* dan *opencv* yang mendukung pembuatan program ini.

c. VRML

Salah satu bahasa komputer untuk membuat model objek 3 dimensi dalam dunia virtual.

✓ PEMBUATAN

1 Mengaktifkan Webcam

Tahap awal yang akan dilakukan oleh program adalah mengaktifkan webcam yang terintegrasi pada komputer, dalam hal ini masih digunakan *webcam internal* dari *netbook* dengan ukuran 640x480 pixel. Ukuran ini dipilih karena merupakan ukuran *standard* dalam pengambilan citra supaya nantinya

proses pencarian obyek tangan menjadi lebih cepat karena resolusi yang digunakan tidak terlalu besar. pembuatan desain dan penyusunan kode, dan tahap pengujian.

2 Pembentukan *Rectangle* (ROI) pada Window

Pembentukan *rectangle* ini akan selalu *ter-display* pada aplikasi ini selama *camera* berada pada posisi menyala atau ON. Sehingga dapat dikatakan bahwa proses ini terkesan berjalan secara bersama-sama, walaupun sebenarnya, proses diawali dari pengaktifan *camera*. Pembuatan *rectangle* pada window aplikasi dimaksudkan untuk mempermudah *user* untuk meletakkan obyek tangan/jari dengan harapan bahwa obyek yang *ter-capture* adalah obyek yang berada didalam *rectangle*. Dimana besar luasan dari *rectangle* ini sama dengan luasan ROI yang akan dibuat. 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. Dan untuk membuat sebuah *rectangle* dilakukan dengan menggabungkan empat koordinasi untuk keempat tepi ROI, yaitu "p1.x ; p1.y ; p2.x ; p2.y".

3. Proses *Learning Background*

Proses *learning background* ini 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*.

4. Proses Perubahan Format RGB menjadi YUV

Proses perubahan format kedalam bentuk *YUV Image* ini dilakukan ke beberapa tahapan proses pembuatan aplikasi ini. Sebut saja seperti pada proses *learning background*. Tetapi, perubahan format ini lebih penting digunakan sebelum menentukan obyek (*foreground*).

```
for(;;)
{
    rawImage = cvQueryFrame( video_capture
);
    if(!rawImage) { break; }
```

```

        cvCvtColor( rawImage, yuvImage,
CV_BGR2YCrCb );/
        auxFindForeground();
}

```

Dari program diatas, terlihat bahwa sebelum melakukan pengecekan obyek (*foreground*), dilakukan terlebih dahulu perubahan format RGB menjadi YUV dengan menggunakan salah satu syntax OpenCV, yaitu "*cvCvtColor(rawImage,yuvImage,CV_BRG2YCrCb.*". Maksud dari sebaris syntax tersebut yaitu dilakukannya perubahan format dari variable gambar *rawImage* yang berupa RGB dan menggubahnya menjadi YUV dan kemudian menyimpannya kedalam variable *yuvImage*.

5. Proses Pendeteksian Obyek (*Foreground*) dalam ROI

Bahwa setelah melakukan konversi format warna, maka proses akan dilanjutkan menuju fungsi *auxfindforeground()*. Dimana fungsi ini akan berisi program atau syntax mengenai pengenalan *foreground* atau obyek yang bukan bagian dari inialisasi *background* pada proses sebelumnya.

6. Proses Pendeteksian *Contour*

Proses pengambilan *contour* dengan salah satu perintah dari openCV, yaitu *cvFindContour*. 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

7. Proses Penentuan Nilai *HuMoments*

Penentuan nilai *HuMoments* ini dilakukan sebagai nilai pembandingan untuk nilai *HuMoments* berikutnya, atau dapat dikatakan bahwa nilai ini merupakan nilai kode *database* dari masing-masing bentuk tangan (gambar).

8. Proses Membandingkan Nilai *HuMoments*

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. Dimana hal ini dimaksudkan untuk memperkecil adanya nilai yang sama didalam rentang nilai yang berbeda antara *gesture* tangan yang satu dengan tangan yang lainnya.

9. Proses Pembuatan Animasi Manusia 3D Dengan VRML

VRML adalah salah satu bahasa computer untuk membuat model objek 3 dimensi dalam dunia virtual. Dunia VRML yang terbangun dari struktur yang terhirarki. File VRML sendiri adalah hirarki yang paling atas, yang terdiri dari header, scene-graph,

prototipe dan event routing. Struktur yang paling bawah adalah node, yang terdiri dari field yang berisi properti dan informasi tentang node yang memilikinya. Kumpulan dari node akan membangun scene-graph. Node yang berisi field juga bisa berisi node lainnya sehingga dapat membentuk sistem kaskading. Header digunakan untuk mengidentifikasi jenis encoding yang dipakai dalam membangun file VRML. Dalam proses pembuatan animasi ini hanya menjelaskan bagaimana animasi bisa terlihat berjalan ke kiri, berjalan ke kanan, berjalan maju dan berjalan mundur. Untuk memperoleh tampilan tersebut maka harus diatur translasi, rotasi dan viewpointnya. Berikut adalah proses translasi untuk animasi supaya berjalan ke kanan :

```

}
DEF jalan Transform { translation 2 0 0
scale 1.4 1.4 1.4
rotation 0 1 0 1.57
.....
]
}

```

Berikut contoh *viewpoint* sintak animasi 3D berjalan ke kiri.

```

.....
viewpoints [
DEF SideView Viewpoint {
description "Side View"
orientation 0 1 0 1.57079
position 2.5929 0.854 0
...
}
]

```

```

void kiri()
{
    CvCapture* capture1 =
cvCreateFileCapture( "jln-kiri.avi" );
    IplImage* frame1;
    while(1) {
        frame1 = cvQueryFrame(capture1);
        //if( !frame1) break;

        cvShowImage("Example2",frame1);
        char c = cvWaitKey(33);
        if( c == 27) break;
    }
}

```

Sintak diatas adalah sintak untuk menjalankan file video .AVI di opencv.

10. Cara Kerja Aplikasi

Cara kerja aplikasi ini, adalah saat webcam menangkap gerakan tangan kita maka isyarat bentuk tangan yang telah di tangkap tadi di program untuk menjalankan aplikasi animasi. Misalnya webcam menangkap bentuk tangan "gun" maka akan

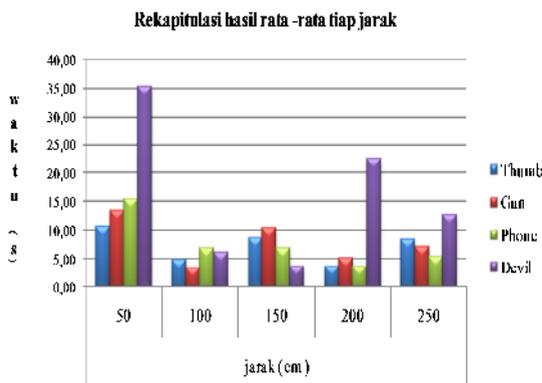
menampilkan aplikasi animasi manusia yang berjalan ke kiri.



Gambar 1. *Software* aplikasi saat di jalankan Begitu seterusnya karena ada empat macam bentuk isyarat tangan maka juga akan menampilkan empat macam gaya aplikasi manusia.

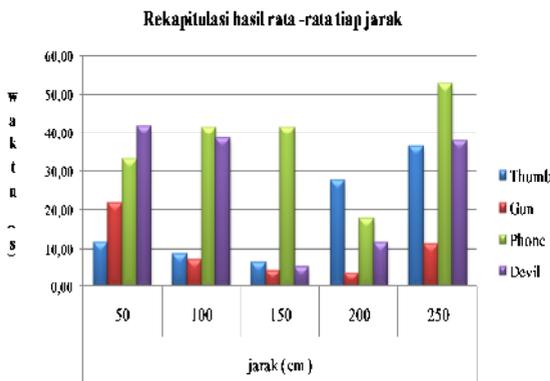
D. Pengujian dan Analisa

- **Pengujian Terhadap Jarak Pada Malam Hari Dalam Ruangn Dengan Intensitas Cahaya 450 lx**



Gambar 2. Grafik Rekapitulasi Rata-Rata Tiap Jarak Malam Hari

- **Pengujian Terhadap Jarak Pada Siang Hari Dalam Ruangn Dengan Intensitas Cahaya 280 lx**

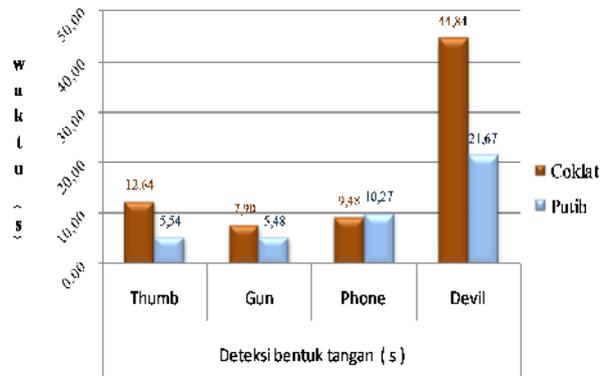


Gambar 3. Grafik Rekapitulasi Rata-Rata Tiap Jarak

- **Pengujian Terhadap Perbedaan Background (jarak 100cm) Dalam**

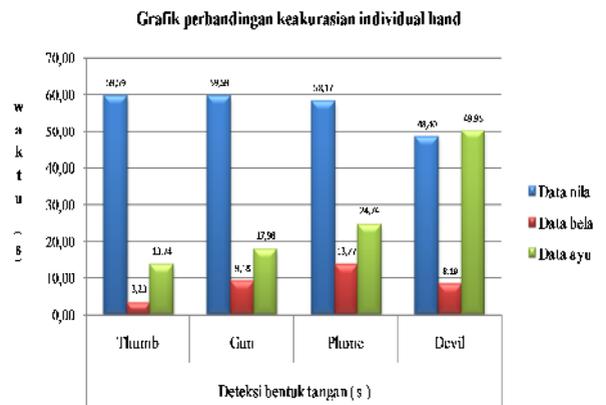
Ruangan Dengan Intensitas Cahaya 255 lx

Grafik respon dari warna background



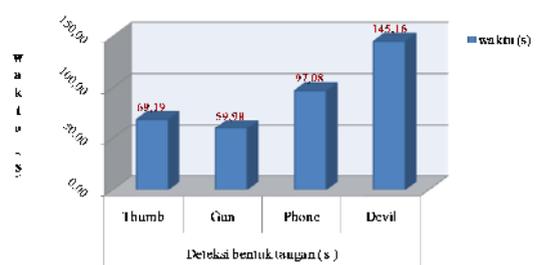
Gambar 4. Perbandingan Antara Background Coklat dan Putih

- **Pengujian Terhadap Perbedaan Tiga Bentuk Tangan Manusia (jarak 100cm) Dalam Ruangn Dengan Intensitas Cahaya 255 lx**



Gambar 5. Perbandingan Perbedaan Bentuk Tangan Manusia

Grafik rekap kecepatan respon berdasar dari semua percobaan



Gambar 6. Grafik Rekapitulasi

Setelah dilakukan beberapa tahap pengujian terhadap *software* meliputi pengujian terhadap jarak dan intensitas cahaya, pengujian jterhadap perbedaan *background* warna coklat dan putih, serta pengujian

terhadap perbedaan tiga perbedaan bentuk tangan manusia dapat diketahui bahwa bentuk tangan yang paling cepat terdeteksi adalah bentuk tangan “gun” dengan rata-rata waktu 59.98(s) sedangkan yang paling lama terdeteksi adalah bentuk tangan “Devil” dengan rata-rata waktu 1:45.16(s).

E. Kesimpulan

Berdasarkan analisa dari beberapa pengambilan data pada bab sebelumnya, maka dapat diperoleh kesimpulan sebagai berikut :

1. Untuk mendapatkan hasil yang optimal dalam pendeteksian tangan maka posisi tangan harus dalam keadaan tegak dan sesuai dengan bentuk tangan yang telah ditentukan.
2. System mampu mengenali obyek tangan dengan baik jika tangan atau obyek dalam kondisi diam atau bergerak lambat.
3. Untuk proses deteksi obyek cukup baik jika di area yang terang.
4. Obyek dapat terdeteksi dengan baik pada latar belakang putih dan latar belakang tidak bergerak seperti dinding atau pintu.
5. Intensitas cahaya 640 lx lebih baik mendeteksi obyek daripada intensitas cahaya 225 lx.

F. Daftar Pustaka

- [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] Gady Agam, “Introduction to Programming with OpenCV”, Departement of Computer Science, 2006
- [3] Mgr. Jan Kapoun, “Static Hand Gesture Recognition Software”, University of South Bohemia, eské Bud jovice, 2010
- [4] Paul Viola and Michaels J.Jones. “*Rapid Object Detection using boosted Cascade of Simple Features*”. IEEE CVPR,2001.
- [5] Open Computer Vision library <http://sourceforge.net/project/opencv/library>
- [6] Ardiansyah Rizky. “*Kendali game Tetris Menggunakan Gerakan Tangan*”, EEPIS-ITS.2009.
- [7] Yusuf. “*Pembuatan Game Puzzle Gambar Dengan Gerakan Tangan dan Perintah Suara*”.EEPIS-ITS.2010