
Rancang Bangun Pendeteksian Gerakan Tangan

Heriansyah*¹, Febriani², Willy³

^{1,2,3}STMIK Global Informatika MDP Jl. Rajawali No.14 Palembang

^{1,2,3}PS Teknik Informatika STMIK Global Informatika MDP

e-mail: *¹heri_ansyah1992@yahoo.com, ²chika.sugoi@gmail.com, ³willy@mdp.ac.id

Abstrak

Pendeteksian menggunakan gerakan tangan telah diaplikasikan secara luas untuk berbagai keperluan, diantaranya pada bidang hiburan, pendidikan serta keamanan. Pengenalan tangan dilakukan dengan kamera (webcam). Kamera ini digunakan sebagai sensor untuk mendeteksi pergerakan tangan. Pendeteksian gerakan tangan diimplementasikan dengan menggunakan algoritme camshift. Algoritme camshift bekerja pada search window yang dapat menemukan pergerakan tangan pada tiap frame. Algoritme camshift yang sudah diterapkan dapat mengkalkulasi ukuran dan lokasi pada search window yang akan digunakan untuk frame selanjutnya. Algoritme camshift sangat baik digunakan untuk pendeteksian seperti pendeteksian tangan ini. Distribusi yang digunakan hue dalam dimensi warna HSV (Hue, Saturation, Value). Penggunaan distribusi hue ini dilakukan untuk mengatasi perbedaan warna kulit manusia dan latar belakang yang digunakan pada saat pengambilan frame. Pada hasil akhir, disimpulkan bahwa algoritme Camshift dapat diterapkan dalam mendeteksi gerakan tangan dengan indikator posisi koordinat x dan y.

Kata kunci—Pendeteksian, Algoritme, Camshift

Abstract

Detection on hand movement have been widely applied for various purposes. Some of them are on entertainment, education and security. The hand recognition is being done using web camera. This camera is used as sensors to capture hand movement. Hand Movement detection used camshift algorithm. Camshift algorithm works on the search window to detect the hand movement in each frame. Camshift algorithm can calculate size and location on search window which would be executed on the next step. Camshift algorithm is rapidly good used for detection, especially on hand object. Hue distribution is used for resolving the disparity between human skin color and background which involved while frame processing.

Keywords—Detection, Algorithm, Camshift

1. PENDAHULUAN

Perkembangan teknologi saat ini sudah sangat pesat, dimana kemajuan teknologi tidak dapat dihindari. Kemajuan teknologi akan berjalan sesuai dengan ilmu pengetahuan. Perkembangan teknologi memang sangat diperlukan setiap manusia. Setiap perubahan yang telah diciptakan sangat memberikan banyak kemudahan bagi manusia dalam melakukan aktivitas.

Komputer dalam pengoperasiannya umumnya menggunakan *hardware* atau alat bantu tertentu untuk melakukan peng-*input*-an perintah pada komputer misalnya *mouse*, *keyboard* dan sebagainya. Pada zaman yang sudah modern ini orang-orang ingin mencoba hal yang baru dan merasa seperti komputer dalam peng-*input*-an perintah.

Hal ini dapat diwujudkan dengan adanya peranan *user interface*. *User interface* merupakan bentuk tampilan grafis yang membantu *user* dapat saling terhubung dengan metode yang dilakukan pada *user interface* kali ini yaitu dengan menggunakan tangan dengan menemukan titik tengah (*centroid*) yang menjadi titik acuan pada telapak tangan dalam proses *tracking* dan mengubahnya dari 3D menjadi 2D pada video. Proses *tracking* yang dapat dilakukan memiliki banyak metode atau algoritme diantaranya yaitu Algoritme *Camshift*. Algoritme *Camshift* ini melacak objek tiap *frame* secara berulang untuk dapat mendeteksi pergerakan objek dari citra video dimana distribusi warna dari video selalu berubah tiap waktu.

2. LANDASAN TEORI

2.1 Kecerdasan Buatan

Alan Turing[1], kecerdasan buatan memiliki pengertian jika sebuah komputer tidak dapat dibedakan dengan manusia saat berbincang melalui terminal komputer, maka bisa dikatakan komputer itu cerdas, mempunyai kecerdasan.

Herbert Alexander Simon[2], kecerdasan buatan merupakan kawasan penelitian, aplikasi dan intruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas.

Rich and Knight[3], kecerdasan buatan merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.

T.Sutojo, S.Si, M. Kom, dkk[4], kecerdasan buatan merujuk pada mesin yang mampu berpikir, menimbang tindakan yang akan diambil dan mampu mengambil keputusan seperti yang dilakukan oleh manusia.

2.2 Citra

Menurut Sutoyo[5], citra adalah suatu representasi (gambaran), kemiripan atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa video yang dapat disimpan pada suatu media penyimpanan. Ada beberapa jenis format citra yang sering digunakan dalam pengolahan citra. Berikut adalah jenis-jenis format *file* citra:

1. *Bitmap* (.BMP)

Format .bmp adalah format penyimpanan standar tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan sebuah nilai *pixel*. Format BMP mendukung RGB, *Indexed Colour*, *Grayscale* dan *Bitmap colour mode*, tetapi tidak mendukung *alpha channel*.

2. *Tagged Image Format* (.TIF, .TIFF)

Format .TIF merupakan format penyimpanan citra yang dapat digunakan untuk menyimpan citra *bitmap* hingga citra dengan warna palet terkompresi. Format ini dapat digunakan untuk menyimpan citra yang tidak terkompresi dan juga citra terkompresi.

3. *Portable Network Graphics* (.PNG)

Format .PNG adalah format penyimpanan citra terkompresi. Format ini dapat digunakan pada citra *grayscale*, citra dengan palet warna dan juga citra *fullcolor*. Format .PNG juga mampu menyimpan informasi hingga kanal *alpha* dengan penyimpanan sebesar 1 hingga 16 bit per kanal. PNG mempunyai kemampuan menampilkan gambar 24 bit dan menghasilkan latar belakang secara transparan.

4. *Joint Photographic Experts Group* (.JPEG , .JPG)

Ekstensi .JPG adalah format yang sangat umum digunakan saat ini khususnya untuk transmisi citra. Format ini digunakan untuk menyimpan citra hasil kompresi dengan metode JPEG.

2.3 *Computer Vision*

Beberapa penelitian yang berhubungan dengan *Computer Vision* telah dilakukan[6][7]. *Computer vision* didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali objek yang diamati atau diobservasi. *Computer vision* menjadikan komputer *Acts Like Human Sight*, sehingga mendekati kemampuan manusia dalam menangkap informasi visual. Untuk mendukung proses ini, maka dalam *computer vision* dilakukan dalam 4 (empat) tahapan utama yaitu :

1. *Image Acquisition*

Akuisisi citra merupakan proses penangkapan informasi visual dan proses pengubahan sinyal analog menjadi data digital yang siap untuk diproses oleh komputer.

2. *Image Processing*

Pemrosesan citra merupakan proses yang memiliki *input* dan *output*-nya berupa citra. Suatu citra ditransformasikan ke bentuk citra lain yang kualitasnya lebih baik.

3. *Image Analysis*

Analisis citra merupakan proses yang memiliki *input* berupa citra dengan *output* bukan citra akan tetapi berupa hasil pengukuran terhadap citra tersebut.

4. *Image Understanding*

Pemahaman citra merupakan proses yang memiliki *input* berupa citra dengan *output*-nya adalah deskripsi tingkat tinggi dari citra tersebut.

2.4 *Pengolahan Citra (Image Processing)*

Menurut T. Sutoyo, S. Si, M. Kom [5], citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat disimpan pada suatu media penyimpanan.

Menurut Rinaldi Munir[6], citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi) yang memegang peranan sangat penting sebagai bentuk visual kaya informasi. Meskipun sebuah citra kaya informasi namun seringkali citra mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (*noise*) warnanya terlalu kontras, kurang tajam, kabur (*blurring*) dan sebagainya.

Agar citra yang mengalami gangguan mudah diinterpretasi baik oleh manusia maupun mesin, maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik.

2.5 *Video*

Video dikatakan sebagai gabungan gambar-gambar mati yang dibaca berurutan dalam suatu waktu dengan kecepatan tertentu. Gambar-gambar yang digabung tersebut dinamakan *frame* dan kecepatan pembacaan gambar disebut dengan *frame rate* dengan satuan fps (*frame per second*). Karena dimainkan dalam kecepatan tinggi maka tercipta ilusi gerak yang halus, semakin besar nilai *frame* rata maka akan semakin halus pergerakan yang akan ditampilkan.

Menurut Jubilee[7], Ada beberapa jenis format video yang biasa digunakan dalam pengolahan citra. Adapun beberapa format video diantaranya :

1. AVI (*Audio Video Interleave*)

Format ini termasuk format video yang tidak dikompresi. AVI adalah format standar *file* video untuk *Microsoft Windows* yang juga merupakan format video tertua karena diperkenalkan sejak *Windows 3.1*. Video yang menggunakan format ini akan menghasilkan ukuran *file* yang sangat besar karena resolusi yang dipakai sesuai dengan resolusi asli dari sumber videonya, yaitu kaset video. Format ini merupakan salah satu format yang berkualitas tinggi karena mampu menghasilkan pergerakan 15 *frame* perdetik dalam resolusi maksimal dengan kualitas suara mencapai 11,025 Hz.

2. *Moving Picture Experts Group* (MPEG)

Format ini merupakan standar untuk hasil kompresi *file* digital video audio. MPEG menghasilkan kualitas gambar yang tinggi tapi tidak membutuhkan kapasitas *file* besar. Kompresi file MPEG terkadang menghilangkan sejumlah *frame* perpindahan sehingga proses transisinya sering tidak enak dipandang. Resolusi video yang berformat MPEG mendukung resolusi setengah layar dan satu layar tergantung jenis MPEG-nya. Format ini memiliki beragam standar antara lain MPEG-1, MPEG-2 dan MPEG-4. Versi MPEG-1 hanya mampu menghasilkan kualitas video dibawah video VCR. Sedangkan untuk MPEG-2 mempunyai kecepatan 60 *frame* per detik sehingga mampu mengompresi video berdurasi 2 jam dalam beberapa *gigabytes* saja. Format MPEG-4 dikeluarkan pada saat tahun 1998 dan biasa digunakan dalam aplikasi internet, ponsel dan televisi. Dengan kapasitasnya yang kecil, MPEG-4 dapat menunjang transmisi via jaringan ber-*bandwidth* kecil. Pada umumnya, kamera *video digital* sudah mampu menghasilkan *output* berupa MPEG.

3. MOV

Termasuk dalam format video terkompresi. MOV dibuat oleh Apple Computer dan dijalankan pada *platform* Macintosh tetapi sekarang dapat dijalankan pada Windows dengan meng-*install* codec Quicktime. MOV termasuk video yang ditujukan untuk *online* video, *website* yang berbasis multimedia dan CD-ROM. Format ini bisa langsung dihasilkan oleh ponsel berkamera yang memiliki fitur membuat video. Salah satu keunggulan format MOV adalah mampu mendukung video interaktif, yaitu *Virtual Reality* (VR).

2.6 *Algoritme Mean Shift*

Algoritme mean shift beroperasi pada citra berwarna pada video. Untuk melacak citra berwarna pada video tersebut harus dipresentasikan dalam bentuk distribusi histogram dari citra dengan menggunakan distribusi histogram dari citra tersebut. Tahap-tahap pada *algoritme mean shift* dapat dijabarkan sebagai berikut :

1. Ukuran *search window* yang sudah ditentukan.
2. Lokasi awal *search window* yang sudah ditentukan.
3. Hitung daerah *mean* dalam *search window*.
4. Posisikan *search window* ke tengah daerah *mean* seperti dihitung pada langkah (3).
5. Ulangi langkah (3) dan (4) hingga konvergen (atau hingga pergeseran daerah *mean* kurang dari *threshold*/ batas yang ditentukan).

2.7 *Algoritme Camshift*

Camshift merupakan singkatan dari *Continuously Adaptive Mean Shift*, yang merupakan pengembangan *algoritme mean shift* yang dilakukan secara terus menerus untuk melakukan adaptasi atau penyesuaian terhadap distribusi probabilitas warna yang selalu berubah tiap pergantian *frame* dari *video sequence*.

Tahap pertama dari *algoritme camshift* adalah dengan melakukan perhitungan terhadap nilai *hue* dari objek sampel warna kulit manusia. Nilai histogram *hue* sebagai hasil perhitungan tersebut selanjutnya akan disimpan untuk digunakan sebagai *lookup table*. *Lookup table* histogram tersebut dapat digunakan untuk semua orang dari berbagai ras (kecuali albino). Perbedaan warna ras yang ada disebabkan karena perbedaan *saturation*. Karena *lookup table*

histogram memiliki nilai *hue*, maka tidak diperlukan *lookup table* baru jika citra yang akan dibandingkan mempunyai warna yang berbeda.

Tahap kedua, dilakukan pemilihan lokasi awal dari *mean shift*. Kemudian dilakukan perhitungan nilai *histogram hue* yang menjadi nilai probabilitas untuk tiap-tiap *pixel* pada citra.

Tahap ketiga, dijalankan algoritma *mean shift* untuk mencari pusat dan besar *search window* yang baru. Lokasi tengah dan besar daerah yang dihasilkan disimpan. Lalu dilakukan *looping* ke tahap dua dimana lokasi awal dan *search window* menggunakan hasil algoritme *mean shift* pada tahap tiga.

Berikut adalah langkah-langkah dari Algoritme Camshift sebagai berikut:

- Langkah 1: Tentukan ukuran awal *search window*
- Langkah 2: Tentukan lokasi awal dari *search window*
- Langkah 3: Tentukan daerah kalkulasi (*calculation region*) pada bagian tengah *search windows* dengan ukuran lebih besar dari *search window*, daerah kalkulasinya, yaitu ROI (*Region of Interest*)
- Langkah 4: *Frame* citra video dikonversi ke dalam sistem warna HSV (*Hue, Saturation, Value*), kemudian langkah selanjutnya membuat histogram dari citra untuk mengetahui distribusi probabilitas warna.
- Langkah 5: Lakukan algoritme *mean shift* (satu atau banyak iterasi) dengan input berupa ukuran dan lokasi *search window* serta citra distribusi probabilitas warna dan simpan *zeroth moment*.
- Langkah 6: Set nilai *x, y, z* yang diperoleh dari langkah (5).
- Langkah 7: Nilai *x* dan *y* dipakai untuk menentukan titik tengah *search window*, sedangkan ($2 \cdot \text{area}^{1/2}$) untuk menentukan ukuran *search window*.
- Langkah 8: Ulangi langkah (3) untuk setiap pergantian *frame* citra video. Untuk citra berdistribusi probabilitas warna, daerah *mean (centroid)* di dalam *search window* bisa dicari dengan persamaan:

Zeroth moment bisa dicari dengan persamaan (1).

$$M_{00} = \sum_x \sum_y I(x, y) \quad (1)$$

First moment untuk *x* dan *y* bisa dicari dengan persamaan (2) dan (3).

$$M_{10} = \sum_x \sum_y x I(x, y) \quad (2)$$

$$M_{01} = \sum_x \sum_y y I(x, y) \quad (3)$$

Maka lokasi *mean* dalam *search window (centroid)* adalah:

$$X_c = \frac{M_{10}}{M_{00}}$$

$$Y_c = \frac{M_{01}}{M_{00}}$$

3. RANCANGAN ALGORITME DAN PROGRAM

3.1 Lingkungan Pengembangan Program

Setelah mempelajari hal-hal yang berkaitan dengan pengembangan aplikasi ini, dalam bab ini akan dibahas mengenai perancangan sistem. Perancangan ini dibagi menjadi dua bagian, yaitu perangkat lunak dan perangkat keras.

Perangkat lunak yang digunakan dalam pengembangan rancangan aplikasi yang akan dibuat yaitu :

1. *Sistem Operasi*

Untuk membantu dalam analisis pendeteksian gerakan tangan dengan metode camshift, dibutuhkan bantuan dari sebuah sistem operasi, dalam hal ini penulis menggunakan sistem operasi Microsoft Windows 7 Ultimate.

2. *Bahasa Pemrograman*

Dalam pembuatan aplikasi ini penulis menggunakan bahasa pemrograman Matlab (*Matrix Laboratory*) dengan versi R2013a.

Penggunaan perangkat keras yang disarankan dalam penggunaan aplikasi ini, yaitu menggunakan komputer yang memiliki kinerja yang cukup cepat, baik dan memiliki memori yang cukup besar. Dengan menggunakan komputer yang memiliki teknologi yang cukup tertinggal, aplikasi yang dibuat oleh penulis ini masih dapat dijalankan, tetapi tentunya dengan penambahan waktu dalam melakukan proses yang lebih lama daripada komputer dengan spesifikasi yang lebih baik.

3.2 Metodologi Iteratif

Dalam mengembangkan aplikasi ini metodologi yang digunakan adalah metodologi *iteratif*. Pengumpulan data yang digunakan dalam penyusunan laporan dilakukan dengan cara membaca buku-buku atau jurnal ilmiah yang berhubungan dengan deteksi tepi pada citra digital dengan menggunakan metode Camshift. Selain membaca buku penulis juga mencari informasi di internet.

Penggunaan suatu metode pengembangan sistem yang tepat dan disesuaikan dengan permasalahan dan aplikasi yang akan dibangun. Metodologi yang digunakan penulis dalam pengembangan aplikasi ini adalah metodologi iteratif. Metodologi iteratif mengkombinasikan proses-proses pada model air terjun (*waterfall*) dan iteratif pada model *prototype*. Dalam metode ini, terdapat 4 tahap pengembangan sistem, yaitu :

1. *Analisis*

Pada tahap ini dilakukan analisis mengenai perangkat keras, perangkat lunak, serta mengumpulkan data-data ataupun informasi yang dibutuhkan untuk membangun aplikasi perbandingan ataupun informasi yang dibutuhkan untuk membangun aplikasi pendeteksian gerakan tangan menggunakan metode camshift.

2. *Desain*

Pada tahap ini dilakukan perancangan dengan membuat *flowchart* dan tampilan antarmuka pengguna (*user interface*).

3. *Pembuatan Kode Program*

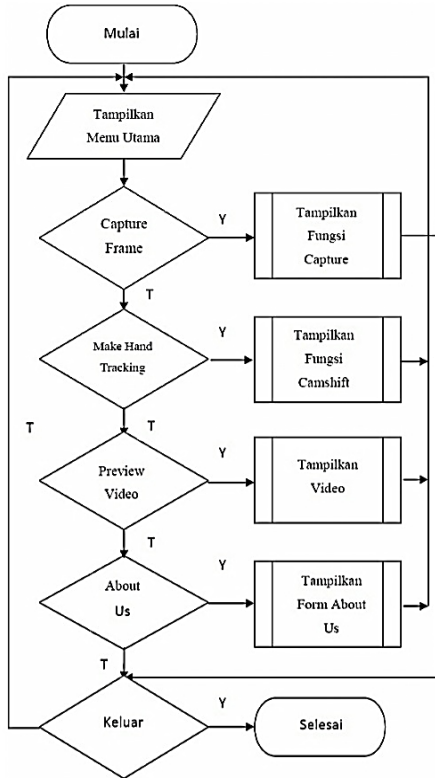
Pada tahap ini akan mengimplementasikan kode program pada rancangan aplikasi yang telah dibuat pada tahap sebelumnya. Bahasa pemrograman yang digunakan adalah Matlab (*Matrix Laboratory*) 2013. Dimana pada tahap ini metode camshift diimplementasikan ke dalam kode program.

4. *Pengujian*

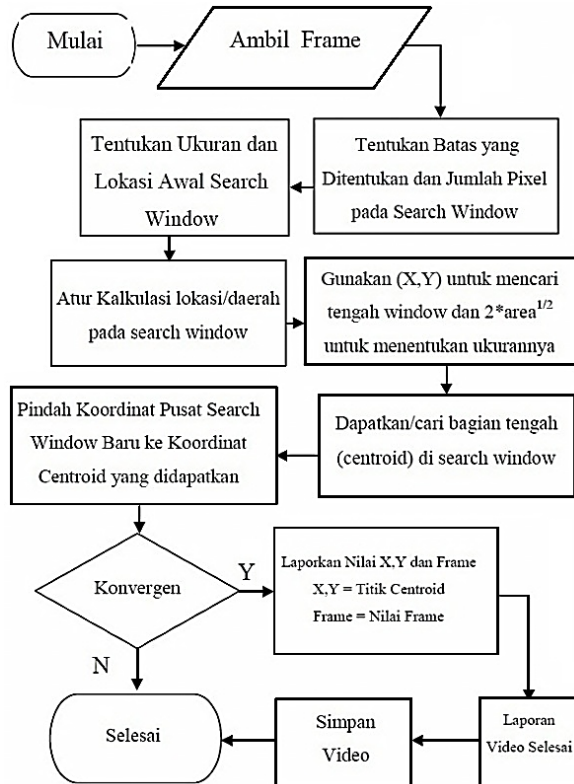
Tahap terakhir yang dilakukan yaitu pengujian serta pemeriksaan semua fungsi-fungsi sistem, apabila terjadi kekurangan ataupun perubahan kebutuhan pada sistem sehingga menyebabkan perlu adanya analisis, desain dan pembuatan kode ulang dengan demikian dilakukan proses pengujian dijalani kembali setelah proses analisis, desain dan pembuatan program.

3.3 Flowchart

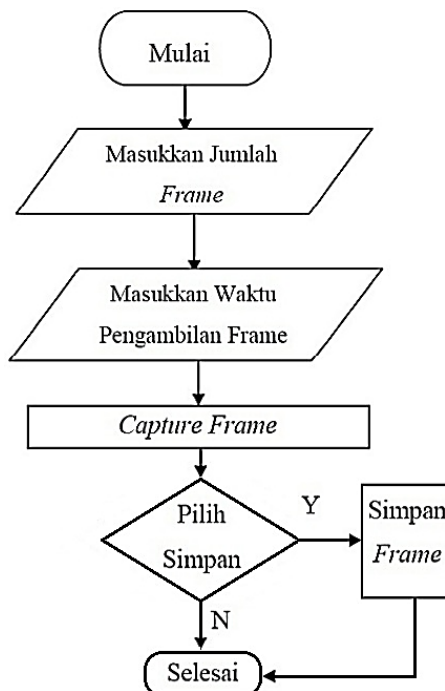
Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program.



Gambar 1 Flowchart Menu Utama



Gambar 2 Flowchart Capture Image



Gambar 3 Flowchart Video Camshift

4. IMPLEMENTASI DAN ANALISIS PROGRAM

4.1 Tampilan Antarmuka Menu Utama

Saat pertama kali menjalankan aplikasi maka *user* akan masuk ke menu utama. Berikut adalah tampilan antarmuka menu utama yang dapat dilihat pada Gambar 4.

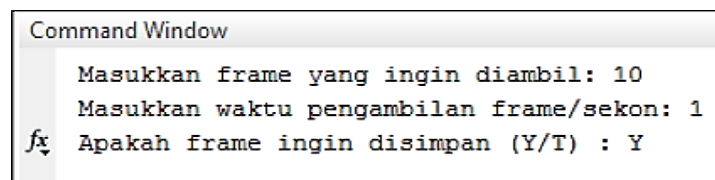


Gambar 4 Antarmuka Menu Utama

Dari menu utama ini akan muncul tampilan yang memberikan pilihan bagi *user*. Pilihan pertama yaitu *capture frame* dimana *user* akan dibawa ke *command window* untuk melakukan peng-input-an jumlah *frame* yang ingin diambil, waktu yg diperlukan untuk pengambilan per *frame* dan *frame* yang diambil ingin disimpan atau tidak. Pada pilihan kedua yaitu *make video hand tracking* pilihan ini dibuat untuk membuat video dari *frame-frame* yang telah diambil. Pilihan ketiga yaitu *preview video* yaitu untuk menampilkan atau memutar hasil video yang telah dibuat. Pilihan keempat yaitu *about us* apabila *user* memilih pilihan ini maka akan ditampilkan profil yang berisi tentang pembuat aplikasi dan yang terakhir yaitu pilihan *exit* digunakan untuk keluar aplikasi.

4.2 Tampilan Antarmuka Capture Image

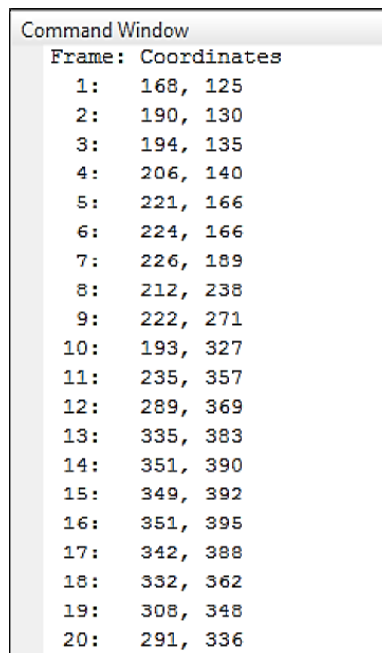
Pada saat *user* memilih *capture image* pada form menu utama, maka akan keluar tampilan *command window* dimana *user* akan mengisi atau meng-input yang diperlukan dalam pengambilan *frame*. Tampilan antarmuka *capture image* dapat dilihat pada Gambar 5.



Gambar 5 Antarmuka Capture Image

4.3 Tampilan Antarmuka Make Hand Tracking Video

Pada saat *user* memilih *make hand tracking video* maka akan ditampilkan pada *command window frame-frame* yang telah diambil beserta koordinat titik untuk *marker/* penanda. Berikut gambar tampilan antarmuka *make hand tracking video* dapat dilihat pada Gambar 6.



```
Command Window
Frame: Coordinates
1: 168, 125
2: 190, 130
3: 194, 135
4: 206, 140
5: 221, 166
6: 224, 166
7: 226, 189
8: 212, 238
9: 222, 271
10: 193, 327
11: 235, 357
12: 289, 369
13: 335, 383
14: 351, 390
15: 349, 392
16: 351, 395
17: 342, 388
18: 332, 362
19: 308, 348
20: 291, 336
```

Gambar 6 Antarmuka *Make Hand Tracking Video*

4. KESIMPULAN

Beberapa kesimpulan yang dapat dibuat dari penelitian ini adalah:

1. Algoritme Camshift sudah dapat diimplementasikan pada aplikasi ini untuk mendeteksi pergerakan tangan, pendeteksian gerakan tangan ini dapat berjalan bergantung pada intensitas cahaya dan latar belakang yang digunakan.
2. Pelacakan objek tiap *frame* dapat dilakukan dengan menentukan koordinat X, Y pada *search window* yang telah ditentukan ukuran serta lokasi *search window* sampai konvergen.

5. SARAN

Berikut adalah saran yang dapat diberikan untuk pengembangan aplikasi pendeteksian gerakan tangan menggunakan algoritme camshift :

1. Mengembangkan aplikasi ini secara *real time*.
2. Mengembangkan aplikasi ini untuk dapat diimplementasikan atau diterapkan ke bidang hiburan, pendidikan dan keamanan.

DAFTAR PUSTAKA

- [1] M. Turing, *Computing Machinery and Intelligence*, Mind, 1950.
- [2] P. Langley, H. A. Simon, G. L. Bradshaw and J. M. Zytkow, *Scientific discovery: Computational Explorations of The Creative Processes*, MA: MIT Press, 1987.
- [3] E. Rich and K. Knight, *Artificial Intelligence*, London: McGraw Hill, 1991.
- [4] T. Sutojo, *Teori Bahasa Otomata dan Komputasi*, Yogyakarta: Andi Offset, 2011.

-
- [5] T. Sutoyo, *Teori Pengolahan Citra Digital*, Yogyakarta: Andi Offset, 2009.
 - [6] T. Ahonen, A. Hadid and M. Pietikinen, "Face Description with Local Binary Patterns: Application to Face Recognition," *PAMI*, p. 28, 2006.
 - [7] Q. Chen, Z. Song, Z. Huang, Y. Hua and S. Yan, "Contextualizing Object Detection and Classification".
 - [8] R. Munir, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Bandung: Informatika, 2004.
 - [9] J. Enterprise, *Teknik Mengubah PC Menjadi Home Theater*, Jakarta: PT Elex Media Computindo, 2010.