

VIMO APPLICATIONS AS VIRTUAL MOUSE WITH REAL TIME GESTURE RECOGNITION

Anisa Lora¹ dan Pipit Dewi Arnesia²

⁽¹⁾Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok, Jawa Barat 16424

⁽²⁾STMIK Jakarta STI&K

Jl. BRI No.17, Radio Dalam Kebayoran Baru Jakarta Selatan 12140

{ansalora86, pdarnesia}@gmail.com

ABSTRAK

Penelitian ini bertujuan untuk mengembangkan suatu aplikasi mouse virtual berbasis desktop untuk mengontrol kursor komputer secara dinamis dengan gerakan tangan yang lebih sederhana. Data yang digunakan dalam penelitian ini berupa citra yang diambil secara real-time melalui webcam dan data latih dari dblib (untuk facial landmark). Setiap citra yang ditangkap oleh webcam akan melewati serangkaian proses metode pengolahan citra untuk mendeteksi kontur tangan dan mata. Gerakan tangan dan mata inilah yang akan digunakan untuk memberikan perintah kursor komputer. Hasil dari penelitian ini menunjukkan bahwa pegeksekusian kode program VIMO untuk menjalankan fungsi kontrol mouse berhasil dijalankan dengan baik dan efektif pada kondisi yang diharapkan. Kondisi latar yang tidak rumit dan kondisi pencahayaan yang tidak terlalu gelap dengan jarak pengguna dan webcam antara 60cm sampai 200cm merupakan kondisi terbaik dalam penggunaan VIMO. Fungsi kontrol yang dapat dilakukan pada aplikasi VIMO yakni kontrol pergerakan kursor, pemberian perintah klik kiri, klik kanan, scroll-up, dan scroll-down, serta fungsi wink untuk kalibrasi latar pengguna.

Kata Kunci : *Computer Vision, Contour Detection, Feature Extraction, Gesture Recognition, Image Processing*

PENDAHULUAN

Penggunaan mouse sebagai *pointing device* untuk mengontrol pergerakan kursor dilayar komputer mengalami banyak perkembangan. Pada masa awal komputer, kontrol kursor hanya dapat dilakukan melalui perintah keyboard. Keterbatasan kontrol kursor melalui keyboard ini kemudian melahirkan konsep *input pointing device*, seperti mouse yang dapat ditambahkan ke sistem komputer sesuai kebutuhan pengguna.

O'Regan (2018) menyebutkan bahwa mouse pertama kali diciptakan oleh Douglas Engelbart pada pertengahan 1960-an^[7]. Mouse ini masih sangat sederhana dan terbuat dari kayu. Hal ini sangat jauh berbeda dengan mouse yang ada pada saat ini, seperti mouse USB dan mouse wireless, bahkan beberapa penelitian telah mengembangkan mouse virtual yang memanfaatkan bagian tubuh manusia untuk berkomunikasi langsung dengan komputer melalui fasilitas web camera atau webcam.

Semua jenis mouse, termasuk mouse virtual, umumnya memiliki fungsi yang

sama, yakni untuk menggerakkan pointer pada layar komputer, namun menurut Haria, Subramanian, Asokkumar, Poddar, dan Nayak (2017), penggunaan pengontrol fisik seperti mouse dan keyboard untuk Interaksi Manusia dan Komputer (HCI) menghindari alami antarmuka, karena ada penghalang yang kuat antara manusia dan komputer^[3]. Rautaray dan Agrawal (2015) juga menyebutkan bahwa, meskipun penemuan keyboard dan mouse adalah kemajuan besar, ada situasi di mana perangkat ini tidak kompatibel untuk HCI^[9]. Solusi efektif untuk masalah ini menurut Mali, Kamble, Gaikwad, Jadhav, Chavan (2018) adalah dengan menerapkan penggunaan isyarat tangan yang dinilai dapat memberikan cara komunikasi yang lebih efektif, menarik dan alami^[5].

Penelitian mengenai teknologi mouse virtual sebelumnya telah banyak dilakukan dengan memanfaatkan berbagai metode. Beberapa diantaranya dapat dilihat pada Gambar 1 berikut ini.

No	Penulis (Tahun)	Judul	Metodologi	Batasan
1	Banerjee, Ghosh, dan Bharadwaj, Saikia (2014)	Mouse Control using a Web Camera based on Colour Detection	- Menggunakan webcam - Metode: <i>Image subtraction algorithm to detect color</i> . - Pergerakan kursor berdasarkan identifikasi warna yang terdeteksi pada jari (merah, hijau, biru)	- Menggunakan alat berupa selotip berwarna - Setiap objek yang memiliki warna (merah, hijau, biru) akan dideteksi sebagai kursor tangan
2	Grif dan Farcas (2015)	Mouse Cursor Control System Based on Hand Gesture	- Menggunakan eksternal kamera - Metode: <i>HSV colorspace, Gaussian filter, erode and dilate morphology</i> - Pergerakan kursor berdasarkan identifikasi warna yang terdeteksi pada jari (merah, hijau, biru)	- Menggunakan alat berupa selotip berwarna - Setiap objek yang memiliki warna (merah, hijau, biru) akan dideteksi sebagai kursor tangan - Gerakan kursor terbatas karena bersifat <i>static gesture</i> - Membutuhkan kamera eksternal
3	Pradhan dan Deepak (2016)	Design of Intangible Interface for Mouseless Computer Handling using Hand Gestures	- Menggunakan webcam - Metode: <i>Computer vision algorithm (color segmentation and feature extraction)</i> - Pergerakan kursor berdasarkan teknik jumlah kursor jari yang terlihat	- Menggunakan alat berupa sarung tangan berwarna merah - Gerakan kursor terbatas karena bersifat <i>static gesture</i>
4	Javed et al., (2018)	Novel Motion Based Cursor Control System, Using Hand Gestures Captured From Accelerometer Through A Motion Detection Technique	- Gerakan tangan diperoleh dengan menggunakan sensor accelerometer ADXL335 - Nilai koordinat dari accelerometer akan menentukan arah pergerakan kursor komputer	- Menggunakan alat (sensor accelerometer ADXL335) - Gerakan kursor terbatas karena bersifat <i>static gesture</i>
5	Mali et al., (2018)	Finger Mouse Movement	- Menggunakan webcam - Metode: <i>Image segmentation using skin color detection, dilation and erosion morphology, and image resize</i> - Pergerakan kursor menggunakan teknik mapping dan gap atau angel	- Sangat bergantung pada intensitas cahaya yang tinggi untuk mendeteksi warna kulit - Sensitiv terhadap objek lain dengan warna sejenis
6	Mazumder et al., (2018)	Finger Gesture Detection and Application Using Hue Saturation Value	- Menggunakan webcam - Metode: <i>HSV colorspace detection, angle and ratio measurement</i> - Pergerakan kursor berdasarkan identifikasi warna yang terdeteksi (merah, biru, hijau)	- Menggunakan alat berupa sarung tangan berwarna - Setiap objek yang memiliki warna merah, biru, atau hijau akan dideteksi sebagai kursor tangan - Gerakan kursor terbatas karena bersifat <i>static gesture</i>
7	Raswan dan Prabowo (2018)	Implementasi Kalman Filter Dalam Teknik Hand Tracking Sebagai Kontrol Pointer Mouse Komputer	- Menggunakan webcam - Metode: <i>Motion Detection, Convexity Defects, Gaussian Filter, Kalman Filtering</i> - Pergerakan kursor berdasarkan teknik jumlah kursor jari yang terlihat	- Sangat bergantung pada intensitas cahaya yang tinggi untuk mendeteksi warna kulit - Sensitiv terhadap objek lain dengan warna sejenis - Gerakan kursor terbatas karena bersifat <i>static gesture</i>

Gambar 1. Penelitian Sebelumnya Tentang Mouse Virtual

Secara umum, penelitian ini mengharuskan pengguna untuk menggunakan alat bantu, menghafalkan warna dan dengan gerakan tangan yang rumit untuk dapat memberikan perintah kursor. Penggunaan pelabelan warna dan

gerakan tangan yang rumit dapat menyebabkan semakin tingginya tingkat kesalahan pemberian perintah oleh pengguna. Selain itu, penelitian sebelumnya juga memiliki gerakan kursor yang terbatas karena bersifat *static gesture*. Pengembangan mouse virtual yang memungkinkan pengguna untuk mengontrol kursor komputer tanpa membutuhkan alat bantu, penghafalan warna, dan gerakan tangan yang lebih sederhana dengan kontrol gerak kursor yang bebas atau bersifat *dynamic gesture* perlu dilakukan agar pemanfaatan mouse virtual lebih optimal dalam menggantikan fungsi perangkat input mouse.

Penelitian ini bertujuan untuk mengembangkan aplikasi mouse virtual berbasis desktop untuk mengontrol kursor komputer secara dinamis dengan gerakan tangan yang lebih sederhana dan tidak memerlukan alat bantu. Aplikasi mouse virtual yang dikembangkan berfokus pada *real time contour detection for gesture recognition using webcam in desktop environment* dengan batasan pada beberapa kondisi:

- Background* pengguna tidak terlalu rumit dan tanpa gangguan objek yang berlebihan
- Pencahayaannya tidak dalam kondisi yang terlalu gelap
- Pengguna menghadap ke webcam dengan jarak antara 60cm sampai 200cm

Dari penelitian ini diharapkan aplikasi mouse virtual yang dikembangkan dapat memberikan manfaat berupa kemudahan bagi pengguna dalam melakukan kontrol kursor pada komputer, sehingga secara perlahan dapat menggantikan fungsi perangkat input untuk menggerakkan kursor seperti mouse.

METODE PENELITIAN

Pada pengembangan Aplikasi Mouse Virtual melalui teknik *real time contour detection for gesture recognition* ini menggunakan metodologi *Modified Waterfall*. *Modified Waterfall Methode* bersifat sekuensial dan terdiri dari lima

tahap yang saling terkait dan mempengaruhi.

Analisis Kebutuhan

Metode yang digunakan dalam pengumpulan data dan informasi pada penelitian ini yakni melalui studi pustaka dengan menghimpun dan menganalisis dokumen, diantaranya penelitian terdahulu, buku, artikel, dan jurnal yang terkait dengan objek penelitian. Data yang digunakan dalam penelitian ini berupa citra yang diambil secara real-time melalui *webcam* dan untuk model data latih *facial landmark*

dari http://dlib.net/files/shape_predictor_68_face_landmarks.dat. Data latih untuk *facial landmark* ini bersifat *opensource*, kecuali untuk kegiatan komersial.

Analisis Sistem

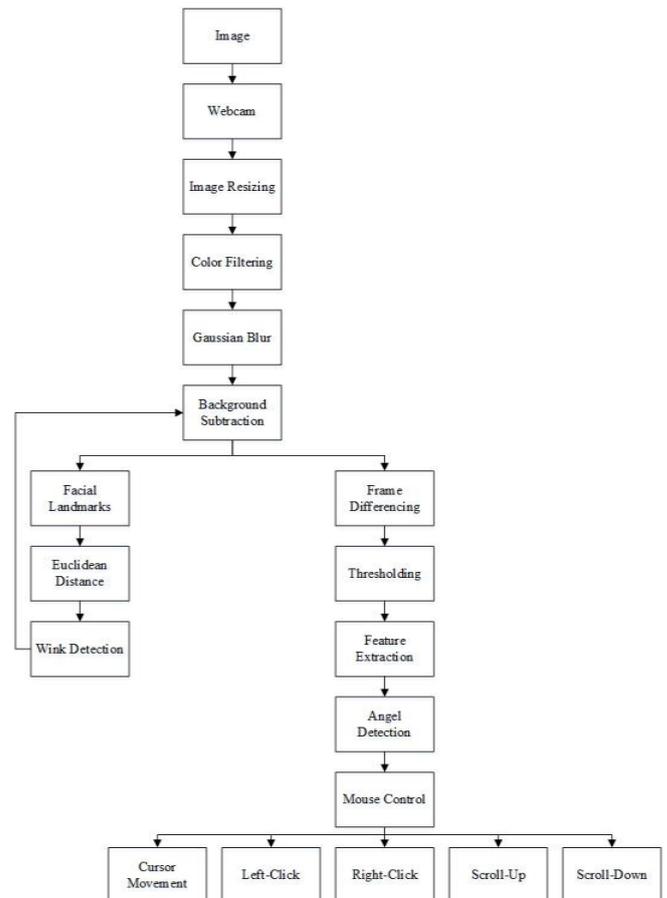
Aplikasi VIMO yang dikembangkan merupakan suatu aplikasi yang berbasis desktop. Aplikasi ini dirancang untuk dapat melakukan kontrol mouse sederhana, seperti menggerakkan pointer, perintah klik kiri dan klik kanan, serta perintah *scroll up* dan *scroll down*. Perintah kontrol mouse ini dapat diproses oleh sistem dengan menangkap informasi gerakan citra oleh *webcam*.

Pada proses pengambilan citra ini, ada beberapa aspek yang perlu diperhatikan, yaitu kondisi latar, pencahayaan, dan jarak citra dengan *webcam*. Setiap citra yang ditangkap oleh *webcam* akan melewati serangkaian proses metode pengolahan citra untuk memanipulasi citra sehingga menghasilkan kualitas citra yang lebih baik. Adapun metode yang digunakan dalam proses tersebut yakni, *Image Resizing*, *Color Filtering*, *Gaussian Blur*, *Background Subtraction*.

Selanjutnya dilakukan deteksi daerah wajah dan kontur tangan melalui metode *Facial Landmarks*, *Frame Differencing*, *Thresholding*, *Feature Extraction* (*contour*). Kemudian, dilanjutkan dengan proses mencari sudut atau angel pada jari tangan dan perhitungan area mata pada wajah. Proses ini dilakukan untuk mendeteksi mata terbuka atau tertutup

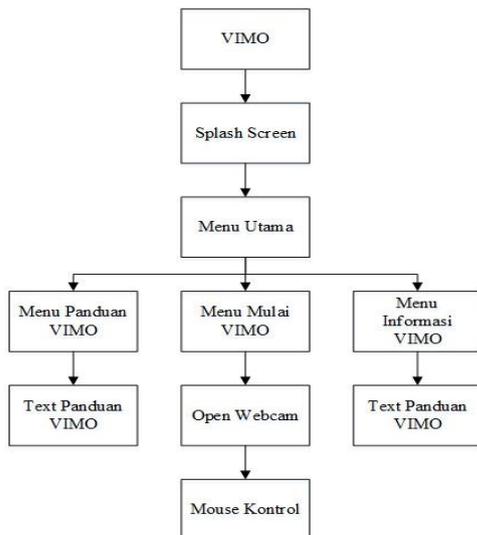
melalui metode *Feature Extraction* (*Convex-Hull* dan *Convexity Defects*) dan *Euclidean Distance*. Angel pada jari tangan dan nilai mata tertutup inilah nantinya yang akan digunakan untuk memberikan perintah kursor pada Aplikasi VIMO.

Adapun arsitektur umum yang dapat menggambarkan metode seperti yang telah dijelaskan sebelumnya ditunjukkan oleh Gambar 2.



Gambar 2. Arsitektur Umum Hand and Face Recognition

Rancangan alur Aplikasi VIMO digambarkan melalui struktur navigasi yang dapat dilihat pada Gambar 3.



Gambar 3. Struktur Navigasi Aplikasi VIMO

Perancangan Tampilan

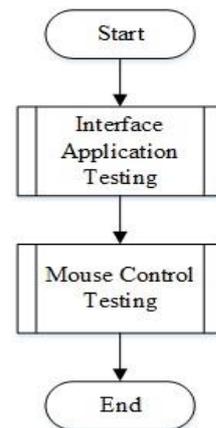
Tahapan perancangan tampilan aplikasi merupakan tindak lanjut dari tahapan analisis. Perancangan ini bertujuan untuk memberikan gambaran mengenai sistem aplikasi yang akan dibangun secara lebih detail.

Implementasi Sistem

Pada tahap ini, rancangan sistem yang telah dibuat sebelumnya diimplementasikan dengan menerapkan serangkaian metode yang telah dijelaskan sebelumnya.

Pengujian Aplikasi

Tahap pengujian ini dilakukan untuk memastikan bahwa sistem bekerja dengan baik seperti yang telah dirancang. Metode yang digunakan dalam pengujian sistem ini menggunakan metode *Black-Box Testing* dengan mekanisme pengujian sebagai berikut:



Gambar 4. Mekanisme Pengujian Aplikasi VIMO

Pengujian ini dilakukan dalam dua tahap, yakni pengujian tampilan aplikasi dan pengujian terhadap kontrol mouse. Pada pengujian tampilan, dilakukan uji terhadap tampilan dan tombol-tombol yang ada pada aplikasi apakah sudah berjalan dengan baik atau masih ditemukan error ketika dijalankan. Pada pengujian kontrol mouse, dilakukan uji input dan output terhadap gerakan yang ditangkap oleh webcam dengan memberikan tiga faktor uji, yakni kondisi latar pengguna, kondisi cahaya ruangan, dan jarak pengguna dengan webcam PC. Pengujian dikatakan berhasil apabila tombol-tombol pada aplikasi dapat menjalankan fungsinya dan kontrol mouse berhasil dilakukan untuk melaksanakan perintah kursor dengan baik

HASIL DAN PEMBAHASAN

Pengembangan Aplikasi Mouse Virtual melalui teknik *real time contour detection for gesture recognition using webcam in desktop environment* ini menggunakan *Library OpenCV* dengan bahasa pemrograman Python. Penelitian ini akan berfokus pada gerakan tangan dinamis. Gerakan tangan diambil menggunakan webcam pada laptop. Gambar yang ditangkap oleh webcam akan diproses melalui serangkaian algoritma yang telah dibangun pada program Python.

Webcam

Proses perolehan citra secara *real time* diambil menggunakan webcam dengan memanfaatkan fungsi *Library OpenCV*

(**cv2.VideoCapture()**). *Webcam* akan terus mengambil gambar agar program dapat terus memproses gambar dan menemukan posisi piksel. Setiap gambar yang diambil akan dilakukan *flipped* menggunakan fungsi (**cv2.flip()**) agar posisi citra asli dengan citra yang tampak dilayar tetap sama atau tidak terbalik.

Resizing

Pada tahap ini dilakukan proses normalisasi ukuran citra yang diambil oleh *webcam* menggunakan fungsi **pyautogui.size()** untuk mengurangi beban proses. Pada penelitian ini, citra yang ditangkap oleh *webcam* memiliki resolusi 1920 x 1080 piksel dan diperkecil menjadi 320 x 220 piksel

Color Filtering

Setelah proses *resizing* selesai, kemudian dilakukan konversi citra ke *grayscale*. Tahap ini disebut dengan proses *grayscale* yang bertujuan untuk mengubah citra yang memiliki komponen warna RGB menjadi komponen warna grey-level. Proses *grayscale* ini dilakukan menggunakan fungsi *library OpenCV cv2.cvtColor()*

Gaussian Blur

Setelah proses *grayscale* selesai dilakukan, maka tahap selanjutnya dilakukan penghalusan citra untuk membantu menghilangkan *noise* dengan menggunakan fungsi **cv2.GaussianBlur()**

Background Subtraction

Tahap selanjutnya dilakukan klasifikasi piksel, apakah itu objek atau *background*. Proses ini dimulai dengan dilakukannya pembacaan frame untuk menentukan frame yang didefinisikan sebagai *background* dan menemukan objek-objek yang ada disekitar *background* menggunakan fungsi *Library OpenCV cv2.accumulateWeighted()*. Frame *background* akan selalu diperbaharui di setiap gambar baru untuk menyesuaikan perubahan lingkungan dari waktu ke waktu. Kecepatan untuk kalibrasi latar, pada penelitian ini diatur sebesar 0.5s. Ketika jumlah frame *background* maksimum telah tercapai, maka dilakukan proses kalibrasi

untuk memperbaharui *background*. Proses ini akan menghasilkan suatu *foreground* dimana objek yang mengganggu disekitar *background* telah terhapus, sehingga pada setiap pembacaan frame berikutnya daerah objek ini tidak akan dibaca.

Frame Differencing

Proses pendeteksian objek bergerak dari urutan frame video yang ditangkap oleh *webcam* menggunakan fungsi *Library OpenCV cv2.absdiff()*.

Thresholding

Proses *thresholding* dilakukan untuk menghasilkan citra biner yang memisahkan objek dari piksel-piksel *noise* hasil perbandingan frame pada tahap sebelumnya. Proses ini hanya memiliki 2 intensitas warna yakni hitam jika nilai piksel citra keabuan lebih kecil atau sama dengan nilai *threshold* dan putih jika nilai piksel citra keabuan lebih besar dari nilai *threshold*. Pada penelitian ini digunakan *Threshold Binary* dengan nilai 25 untuk *threshold* dan 255 untuk *maxVal*. Proses ini terus dilakukan sampai dengan frame terakhir.

Feature Extraction

Tahap ini dilakukan untuk mendapatkan *contour*, *convex hull*, *convexity defect*, dan *center of palm* atau *moment* pada object disetiap frame. Tahap ekstraksi fitur dilakukan dengan membaca frame gambar yang sudah dalam bentuk *grayscale*.

- Fungsi *library OpenCV cv2.findContours()* dapat digunakan untuk menemukan kontur *image*. Berdasarkan area kontur yang terdeteksi, dicarilah area kontur maksimum tangan. Hasil proses deteksi kontur ini digambar menggunakan garis merah
- Temukan *convex-hull* dari wilayah tangan yang disegmentasi (yang merupakan kontur) menggunakan fungsi **cv2.convexhull()**
- Temukan *convexity defects* yang merupakan titik temu antar titik kontur dan garis *convex-hull* untuk menghitung jarak angel tangan menggunakan fungsi *library OpenCV*

cv2.convexityDefects(). Kemudian, *convexity defects* ini diberikan kondisi jika ≤ 50 , maka angel “Merah” dihitung, dan jika > 50 , maka angel “Kuning” dihitung

- Temukan *center of palm* menggunakan fungsi *moment*. *Center of palm* ini nantinya akan menjadi pusat dari pergerakan kursor dan digambarkan melalui lingkaran putih

Face Detection

Tahap ini merupakan proses mendeteksi wajah pada setiap frame dengan cara terus melakukan *looping*. Tahap pertama yang dilakukan adalah menentukan *facial landmark* untuk daerah wajah menggunakan pustaka *dlib* yang telah dijelaskan pada bagian [2]. Program ini menunjukkan bagaimana cara menemukan bagian wajah manusia berdasarkan 68 *landmarks* atau titik pada wajah seperti mulut, alis, dan mata. Pada penelitian ini, hanya bagian mata yang akan digunakan untuk diproses. Tahap selanjutnya dilakukan konversi *facial landmark* (x, y) -coordinates menjadi NumPy array melalui fungsi *predictor*. Koordinat ini nantinya digunakan untuk menghitung EAR kedua mata menggunakan fungsi *euclidean distances*.

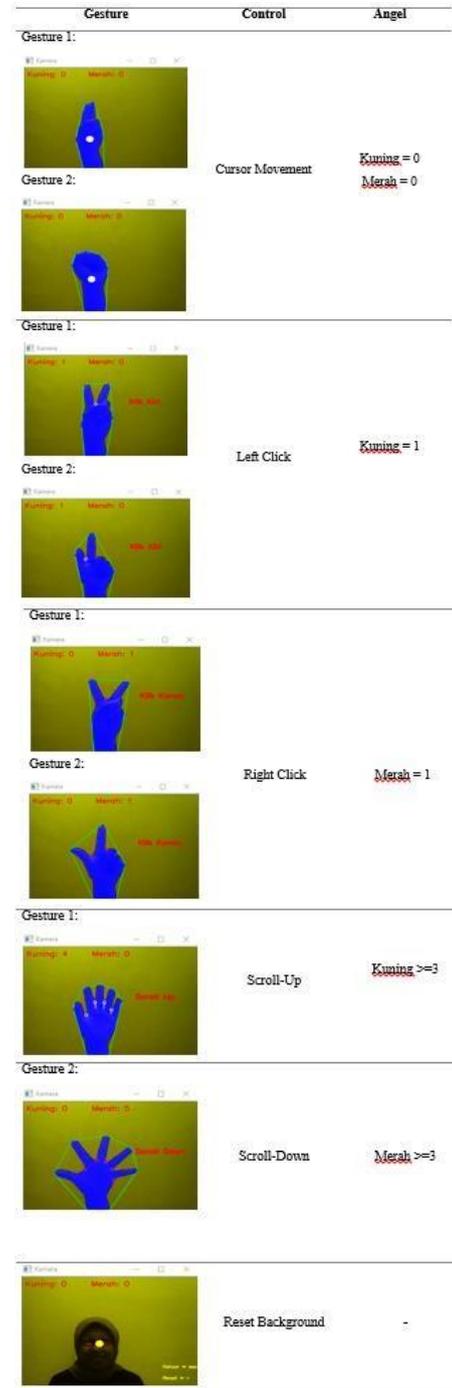
Euclidean Distance

Pada tahap ini dilakukan perhitungan EAR menggunakan fungsi *euclidean distances*. Kemudian dilakukan perhitungan mata tertutup atau *wink detection* pada frame. Mata tertutup terdeteksi dengan kondisi jika nilai ambang EAR kedua mata setelah dibagi dua lebih kecil atau sama dengan nilai ambang EAR dan nilai *frameEAR* lebih kecil dari 30

Gesture Recognition

Program akan mengeksekusi kontrol mouse berdasarkan gerakan yang telah ditentukan menggunakan jumlah angel yang terdeteksi pada tangan. Kontrol mouse dilakukan dengan memanfaatkan Pustaka *PyAutoGUI*. Proses pengenalan *hand gesture* dimulai ketika tangan pengguna diletakkan di depan bidang pandang webcam setelah proses kalibrasi selesai. Apabila wilayah tangan dan wajah

tersegmentasi, maka dilakukan penggambaran daerah tangan dan mata. Gesture pada Aplikasi VIMO dapat dilihat pada Gambar 5 berikut.



Gambar 5. Gesture Aplikasi VIMO

Pengeksekusian kontrol mouse tidak bergantung pada *gesture* jari, akan tetapi berdasarkan sudut jari atau angel yang terdeteksi. Gambar 5 diatas hanya

memberikan gambaran *gesture* yang disarankan untuk melakukan kontrol mouse. Pengguna juga dapat memberikan *gesture* sesuai dengan keinginan sendiri dengan kondisi jumlah angel atau sudut yang terdeteksi sesuai dengan ketentuan.

Pada tahap pengujian Aplikasi VIMO, ada tiga faktor uji yang diberikan, yakni faktor latar pengguna, faktor pencahayaan, dan faktor jarak antara pengguna dan webcam. Dari tiga faktor uji ini, masing-masing faktor kemudian diberikan lagi tiga kondisi dengan rincian sebagai berikut:

1. Faktor Uji Latar Pengguna
 - Sangat Sederhana: Kondisi latar berupa dinding polos dan tanpa ada objek atau benda yang mengganggu
 - Sederhana: Kondisi latar berupa dinding polos, namun terdapat dua atau tiga objek berukuran kecil yang mengganggu
 - Rumit: Kondisi latar dengan banyak objek yang mengganggu.
2. Faktor Uji Pencahayaan Ruang
 - Terlalu Terang: Kondisi pencahayaan terlalu terang, yakni pengguna berada pada ruang terbuka dengan intensitas cahaya sangat tinggi
 - Normal: Kondisi pencahayaan tidak terlalu terang atau terlalu gelap, yakni kondisi ruang belajar atau ruang kelas.
 - Terlalu Gelap: Kondisi pencahayaan terlalu gelap, yakni pengguna berada pada ruangan dengan penerangan yang minim.
3. Faktor Uji Jarak Antara Pengguna dan Webcam
 - <60cm: Jarak antara pengguna dan webcam lebih kecil dari 60cm
 - 60cm sampai 200cm: Jarak antara pengguna dan webcam antara 60cm sampai 200cm
 - >200cm: Jarak antara pengguna dan webcam lebih dari 200cm

Berdasarkan tiga faktor uji beserta kondisi yang telah diberikan, dilakukanlah pengujian dari sisi pengguna. Keakuratan

pada pengujian aplikasi VIMO ini dicari menggunakan perhitungan pada persamaan (1) dan (2) berikut. ^[11]

$$\text{Akurasi} = \frac{\text{Jumlah Testing Berhasil}}{\text{Jumlah Percobaan (Skenario Per Tahap Uji)}}$$

$$\text{Total Akurasi} = \frac{\text{Jumlah Total Akurasi (Q)} \times 100\%}{\text{Jumlah Total Tahap Uji}}$$

Berdasarkan tiga faktor uji yang telah diberikan, didapatlah hasil uji bahwa total akurasi penggunaan VIMO sesuai dengan kondisi yang diharapkan adalah 92,5%. Faktor kondisi latar pengguna yang tidak rumit dan kondisi pencahayaan yang tidak terlalu gelap dengan jarak antara 60cm sampai 200cm merupakan kondisi paling baik dan efektif dalam penggunaan VIMO, dimana semua fungsi kontrol mouse dan fungsi wink berjalan dengan baik.

Hasil untuk tahapan pengujian terhadap faktor lainnya memiliki beberapa kekurangan. Pengujian dengan kondisi latar yang rumit menyebabkan fungsi kontrol mouse dan fungsi wink tidak dapat berjalan dengan baik akibat terlalu banyaknya gangguan objek. Pengujian dengan kondisi pencahayaan terlalu terang menyebabkan fungsi kontrol mouse dan fungsi wink tidak dapat berjalan dengan baik. Hal ini disebabkan karena fungsi wink menjadi terlalu sensitive, sehingga sistem menjadi terlalu sering untuk melakukan proses kalibrasi latar. Pengujian dengan kondisi pencahayaan terlalu gelap juga menyebabkan fungsi kontrol mouse dan fungsi wink tidak dapat dilakukan. Hasil proses pengujian terhadap faktor jarak pengguna dengan webcam yang terlalu jauh menyebabkan fungsi kontrol mouse tidak dapat dilakukan karena sistem tidak dapat mendeteksi kontur tangan dan mata pengguna.

PENUTUP

Kesimpulan

Penelitian untuk pengembangan sistem Aplikasi VIMO sebagai aplikasi mouse virtual dalam mengontrol kursor komputer telah berhasil dilakukan sesuai dengan tahapan penelitian yang telah direncanakan.

Berdasarkan hasil pengujian yang telah dilakukan, didapatkan kesimpulan bahwa secara keseluruhan Aplikasi VIMO dapat berjalan dengan baik. Aplikasi VIMO berhasil menampilkan halaman menu yang telah disediakan melalui fungsi tombol-tombol yang ada pada aplikasi. Pengeksekusi kode program VIMO untuk mengontrol gerakan kursor komputer dan pemberian perintah kursor juga berhasil dilakukan pada kondisi yang diharapkan.

Hasil uji menunjukkan bahwa kursor dapat digerakkan secara dinamis dengan *gesture* yang lebih sederhana dan tanpa menggunakan bantuan alat. Pemberian perintah klik kiri, klik kanan, *scroll-up*, dan *scroll-down*, serta fungsi *wink* untuk kalibrasi latar pengguna berhasil dijalankan dengan baik. Kondisi latar yang tidak rumit dan kondisi pencahayaan yang tidak terlalu gelap dengan jarak pengguna dan webcam antara 60cm sampai 200cm merupakan kondisi dengan hasil uji terbaik dalam penggunaan VIMO. Total Akurasi penggunaan VIMO yang diperoleh dari keseluruhan tahapan pengujian yang dilakukan adalah sebesar 92,5%.

Saran

Berdasarkan besarnya pengaruh faktor latar, pencahayaan, dan jarak antara pengguna dengan sistem, diharapkan pengembangan aplikasi mouse virtual selanjutnya dapat mengurangi atau menghilangkan pengaruh dari faktor tersebut. Penelitian selanjutnya juga disarankan untuk dapat membuat *gesture* yang lebih *user friendly*, sehingga penggunaan mouse virtual dapat lebih efektif lagi untuk menggantikan fungsi perangkat input mouse.

DAFTAR PUSTAKA

- [1] Banerjee, A., A. Ghosh, K. Bharadwaj dan H. Saikia, "Mouse Control Using a Web Camera Based on Colour Detection", *International Journal of Computer Trends and Technology (IJCTT)*, vol 9, no. 1, pp. 15-20. 2014.
- [2] Grif, H. S. dan C. C. Farcas, "Mouse Cursor Control System Based on Hand Gesture", *9th International Conference Interdisciplinarity in Engeneering Romania*, pp. 657 – 661, 8-9 Oktober 2015.
- [3] Haria, A., A. Subramanian, N. Asokkumar, S. Poddar dan J. S. Nayak, "Hand Gesture Recognition for Human Computer Interaction", *7th International Conference on Advances in Computing & Communications*, pp. 367-374, 24 Agustus 2017
- [4] Javed, S., A. Singh, G. Banu, S. Moula, M. Amir dan M. Hassan, "Novel Motion Based Cursor Control System, Using Hand Gestures Captured From Accelerometer Through A Motion Detection Technique", *Global Journal for Research Analysis*, vol. 7, no. 4, pp. 66-69. 2018.
- [5] Mali, S. P., J. B. Kamble, T. L. Gaikwad, B. S. Jadhav dan M. M. Chavan, "Finger Mouse Movement", *International Journal of Engineering and Management Research*, vol. 8, no. 2 pp. 5-7. 2018.
- [6] Mazumder, J., L. N. Nahar dan M. U. Atique, "Finger Gesture Detection and Application Using Hue Saturation Value", *I.J. Image, Graphics and Signal Processing*, pp. 31-38, 8 Agustus 2018.
- [7] O'Regan, G., *The Innovation in Computing Companion*. Switzerland: Springer Nature Switzerland AG. Cham, 2018.
- [8] Pradhan, A., dan B. B. V. L. Deepak, "Design of Intangible Interface for Mouseless Computer Handling using Hand Gestures", *7th International Conference on Communication, Computing and Virtualization*, vol.79, pp. 287-292, 2016.
- [9] Rautaray, S. S. dan A. Agrawal, "Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey", *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.
- [10] Reswan, Y. dan D. A. Prabowo, "Implementasi Kalman Filter Dalam Teknik Hand Tracking Sebagai

- Kontrol Pointer Mouse Komputer”,
Jurnal Sistem Informasi (JSI), vol.
10, no. 1, pp. 1448-1462, 2018.
- [11] Simaremare, H. dan A. Kurniawan,
“Perbandingan Akurasi Pengenalan
Wajah Menggunakan Metode LBPH
dan Eigenface dalam Mengenali Tiga
Wajah Sekaligus secara Real-Time”,
*Journal of Science, Technology and
Industry*, Vol. 14, no. 1, pp. 66-71,
2016.