

Perancangan Artificial Intelligence Hand Tracking menggunakan Algoritma Pyramidal Lucas-Kanade Optical Flow

Rizualdi Fadli¹, Hadi Syaputra², A. Haidar Mirza³, Nia Oktaviani⁴

^{1,2,3,4}Jurusan Teknik Informatika, Ilmu Komputer, Universitas Bina Darma Palembang

Email: rizualdifadli92@gmail.com¹, hadisyaputra@binadarma.ac.id²,

haidar.mirza@binadarma.ac.id³, niaoktaviani@binadarma.ac.id³

Abstrak

Saat ini, Komputasi tidak terbatas pada desktop dan laptop sehingga telah menemukan jalannya ke perangkat seluler seperti Smartphone. Tapi ada juga yang tidak berubah selama 50 tahun terakhir? Seperti beberapa perangkat input, Mouse dan Keyboard. Mouse virtual menggunakan teknologi sensor dan kecerdasan buatan untuk memungkinkan pengguna bekerja di permukaan apa pun seolah-olah itu adalah Mouse. Penelitian ini mengembangkan aplikasi untuk memvisualisasikan Mouse komputer dengan konsep gambar pengolahan. Keyboard virtual harus dapat diakses dan berfungsi. Dengan bantuan gambar kamera Mouse akan diambil. Pergerakan akan ditangkap oleh kamera, saat kita melakukan gerakan yang telah di program maka suatu aksi dapat di tampilkan di dekstop. Kamera akan menangkap gerakan jari saat bergerak. Jadi pada dasarnya ini memberikan Virtual Mouse. Penelitian ini juga menyajikan visi berbasis mouse virtual yang akan mengambil koordinat jari sebagai input. Aplikasi dan kamera akan Bergerak mendeteksi jari kita untuk mengendalikan mouse kita. dengan bantuan kamera sebagai alat pengambil gambarnya.

Kata Kunci: Kecerdasan buatan, Mouse virtual antarmuka, Algoritma Pyramidal Lucas-Kanade Optical Flow

Abstract

Nowadays, Computing is not limited to desktops and laptops so it has invented mobile devices like Smartphones. But there's also something that hasn't changed over the last 50 years? Such as multiple input devices, Mouse and Keyboard. The virtual mouse uses sensor technology and artificial intelligence to allow the user to work on any surface as if it were a mouse. This study develops an application to visualize a computer mouse with the concept of image processing. The virtual keyboard must be accessible and working. With the help of Mouse camera pictures will be taken. The movement will be captured by the camera, when we make a movement that has been programmed then an action can be displayed on the desktop. The camera will capture finger movements as they move. So basically it gives Virtual Mouse. This study also presents a virtual mouse-based vision that will take finger coordinates as input. Applications and cameras will detect our finger to control our mouse. with the help of the camera as a means of taking pictures.

Keywords: Artificial intelligence, Virtual mouse interface, Lucas-Kanade Optical Flow Pyramidal Algorithm

PENDAHULUAN

Dilansir dari aptika.kominfo.go.id Teknologi komputer terutama dalam bidang AI (*Artificial Intelligence*) merupakan sebuah teknologi komputer atau mesin yang memiliki kecerdasan layaknya manusia dan bisa diatur sesuai keinginan manusia. AI (*Artificial Intelligence*) bekerja dengan mempelajari data yang diterima secara berkesinambungan. Semakin banyak data yang diambil, semakin baik pula AI dalam membuat prediksi. Aplikasi chatbot, pergerakan tangan (Hand Gesture) dan pengenalan wajah (face recognition) merupakan salah satu contoh penerapan AI. Komputer telah mengalami miniaturisasi yang cepat dari menjadi 'penghemat ruang' menjadi 'sekecil telapak tangan Anda'. Contohnya, Disk dan komponen lainnya tumbuh lebih kecil dalam ukuran, tetapi satu komponen yang masih tetap sama selama beberapa tahun terakhir salah satu contohnya adalah Mouse. Hand tracking merupakan salah satu implementasi teknik tracking and motion, di mana tracking dilakukan pada objek tangan manusia, dalam hal ini sumber video merupakan video yang dihasilkan oleh webcam secara realtime yang menampilkan objek visual berupa tangan dari seorang individu atau seseorang, selanjutnya akan dilakukan pendeteksian gerakan dari tangan orang tersebut, sehingga komputer atau webcam

dapat membaca dan menerima gerakan arah dari orang tersebut. Dalam implementasi hand tracking terdapat penggunaan metode tertentu yang menjadi dasar atau merupakan inti dari hand tracking, salah satunya yaitu algoritma pyramidal Lucas-Kanade(Karisma Hendri. 2011.).

Algoritma Lucas-Kanade (1981, dalam wahyu 2020) merupakan perpindahan konten gambar antara dua instance terdekat dan konstan dalam lingkungan titik yang dipertimbangkan. Algoritma Lucas-Kanade memperkirakan titik yang dilacak dan titik sebelahnya dari titik yang dilacak memiliki nilai intensitas yang sama. Algoritma Lucas-Kanade memiliki perpindahan posisi titik yang dilacak dari frame ke frame adalah kecil. Posisi, arah gerak, besar pergerakan yang dilakukan dan kecepatan gerak dari titik yang dilacak pada frame dapat diukur. Dalam computer vision, metode Lucas-Kanade adalah metode diferensial yang banyak digunakan untuk estimasi optical flow yang dikembangkan oleh Bruce D. Lucas dan Takeo Kanade(Hendri, 2011). Oleh karena itu Dengan adanya implementasi teknologi visual dalam berinteraksi dengan komputer kemungkinan besar dalam 10 tahun ke depan, akan terjadi perkembangan dalam area yang juga dikenal dengan *Human Computer Interaction*(HCI). mouse bukan lagi primadona dalam berinteraksi dengan komputer. Teknologi HCI seperti *speech recognition* dan *gesture recognition* akan mulai digunakan di mana-mana. Salah satu penerapan teknik interaksi berbasis visi komputer ini adalah pengendalian *pointer mouse* dengan menggunakan webcam. Kamera atau Webcam akan digunakan sebagai sensor untuk menelusuri pergerakan atau gestur tangan. Selanjutnya, gestur tangan ini diterjemahkan dalam aksi mouse.

Pada tahun 2011 oleh Hendri yang mana pada penelitian tersebut juga menggunakan algoritma lucas kanade untuk melakukan pengujian keakuratan terhadap perbedaan warna kulit dengan format YcrCb. Namun kelemahan terjadi pada pemanfaatan warna range kulit dalam format YCrCb karena kemungkinan akan terjadi ketidakstabilan ketika proses tracking kontur yang diciptakan hanya berdasarkan perbedaan warna kulit. Pada tahun 2013, Yuli dkk melakukan pengujian pengendalian pointer menggunakan algoritma lucas kanade dengan pengujian kecepatan gerakan, dan pengujian klik pointer, diperoleh tingkat keberhasilan pada webcam 16 MP sebesar 87,8% sedangkan pada webcam 0,3 MP sebesar 78,9%. Penelitian lain tentang pengenalan gerakan tangan dilakukan pada tahun 2013 oleh Ahmad yang menjelaskan tentang bahwa sistem pengenalan jari tangan manusia pada sistem kamera proyektor dapat digunakan sebagai mouse pointer, akan tetapi sampai saat ini sistem hanya dapat berjalan pada kondisi background yang bernoise rendah atau objek benda yang sedikit.

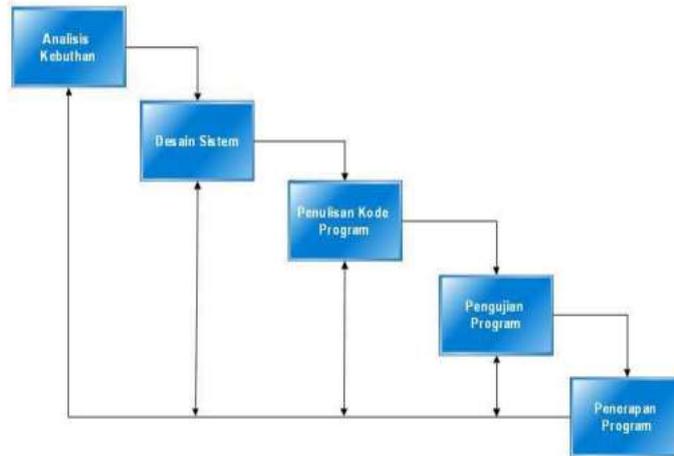
Dari paparan penelitian terdahulu di atas, penulis mencoba untuk membahas bagaimana cara untuk meningkatkan akurasi atau kekurangan tersebut menggunakan algoritma Lucas-Kanade Optical Flow dan di bantu beberapa alat dan *software* yang dapat meningkatkan keakuratan dan membedakan warna kulit terhadap background noise yang tinggi. dalam penelitian ini dilakukan implementasi hand tracking menggunakan algoritma pyramidal lucas-kanade dengan membangun sistem pengontrol pointer mouse pada komputer dengan menggunakan gerakan tangan dari pengguna, dan penelitian ini pun merupakan salah satu implementasi teknologi interaksi berbasis visi komputer yang bersifat alami. Pergerakan tangan mouse akan disesuaikan dengan pergerakan tangan yang di inputkan pada webcam secara *realtime* dan kontrol klik kanan dan klik kiri disesuaikan dengan posisi dari telapak tangan. Tujuan yang diharapkan dari penelitian ini yaitu untuk dapat membantu pengguna dalam berinteraksi dengan komputer untuk melakukan kontrol mouse pointer dengan menggunakan pergerakan alami dari tangan.

Adapun manfaat yang dapat di ambil dari penelitian ini adalah:

1. Dapat memudahkan manusia dalam proses pengetikan dan pergerakan cursor mouse tanpa harus menjaukau area keyboard dan mouse itu berada.
2. Tidak perlu menambahkan perangkat keras (Hardware) lainnya untuk menggunakan AI Virtual (*Artificial Intelligence*) Mouse karena setiap laptop dan smartphone pasti sudah di sediakan kamera.
3. Memanfaatkan dan memaksimalkan fungsi kamera depan smartphone/laptop sebagai media yang untuk menggunakan keyboard/mouse virtual tersebut.

METODE

Pengumpulan data dilakukan untuk memperoleh data yang dibutuhkan dalam pembuatan sebuah program aplikasi. Dalam penelitian ini teknik pengumpulan data yang dilakukan dengan Literatur Review dan sumber dataset menggunakan data sekunder. Metode ini merupakan serangkaian kegiatan yang berkenaan dengan metode pengumpulan data pustaka, membaca dan mencatat, serta mengelola bahan penelitian dari buku maupun jurnal (Zed, 2008:3) yang berkaitan dengan informasi dari pembuatan aplikasi yang dibuat pada tahap analisa data. Dalam membangun sistem ini, digunakan metode pengembangan perangkat lunak dengan menggunakan metode waterfall sebagai berikut :



sumber : medium.com
Gambar 1. 1 Metode Waterfall

a. Analisa kebutuhan

Tujuan dilakukan tahapan ini untuk memahami sistem yang sedang berjalan agar dapat mendefinisikan permasalahan sistem sehingga selanjutnya dapat menentukan kebutuhan sistem secara garis besar sebagai persiapan ke tahapan perancangan. Analisis disini dilakukan dengan pemodelan menggunakan metode pemrograman berorientasi object dengan Tool Unified Modeling Language (UML).

b. Desain sistem

Pada tahap perancangan ini diberikan gambaran umum yang jelas kepada pengguna dan rancang bangun yang lengkap tentang sistem yang akan di kembangkan kepada pihak-pihak yang terlibat dalam pengembangan sistem. Tahap perancangan ini digunakan untuk persiapan implementasi.

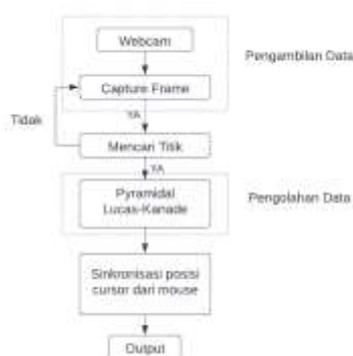
c. Penulisan kode program

Setelah tahap perancangan sistem selanjutnya dilakukan konversi rancangan sistem kedalam kode-kode bahasa pemrograman yang diinginkan Pada tahap ini dilakukan pembuatan komponen-komponen sistem yang meliputi modul program, antarmuka, dan basis data.

d. Pengujian program

Pengujian software dilakukan untuk memastikan bahwa software yang dibuat telah sesuai dengan desainnya dan semua fungsi dapat dipergunakan dengan baik tanpa ada kesalahan.

Langkah-langkah yang dilakukan dalam penelitian ini secara umum dapat dibagi menjadi dua kegiatan utama, yaitu pengambilan data, dan pengolahan data. Pengambilan data adalah pekerjaan yang dilakukan pertama kali. Hasilnya adalah Frame dari gerak tangan. Frame tersebut kemudian menjadi input bagi proses pengolahan data. Alur pekerjaan diperlihatkan pada gambar 2.



Sumber : lucidchart.com
Gambar 1. 2 Flowchart fungsi deteksi jari tangan

1.1 Perangkat Keras

Adapun Spesifikasi perangkat keras yang digunakan pada penelitian ini sebagai berikut:

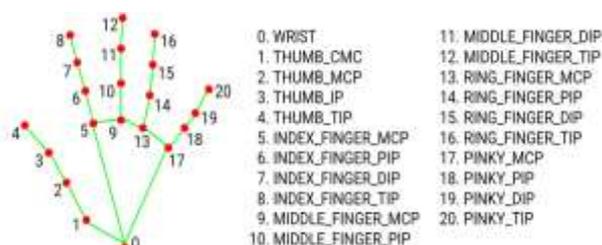
1. *Processor* = Intel Core i5 8th gen 3,4GHz.
2. *Memory* = 8 GB(dua 4 GB) DDR4 2400MHz.
3. *Storage* = SSD 250 GB dan HDD 1 TB.
4. Resolusi Layar = 1366 x 768.
5. Kamera = 2mp dan 6mp.

1.2 OpenCV (Open Computer Vision)

Computer Vision adalah ilmu dan teknologi mesin yang mampu mengekstrak informasi dari gambar yang diperlukan untuk menyelesaikan tugas tertentu(<https://www.intel.co.id/>). Sebagai suatu disiplin ilmu, di balik sistem buatan Computer Vision berkaitan dengan teori yang mengekstrak informasi dari gambar. Data gambar dapat berupa video, pandangan dari beberapa kamera, atau data multi-dimensi dari scanner medis(siti 2017, h.7). Pendapat lain juga di utarakan oleh kurniawan (2017) OpenCV (Open Source Computer Vision) adalah open source library yang berisi lebih dari 500 algoritma teroptimasi untuk analisa citra atau video. Dapat disimpulkan bahwa OpenCV merupakan framework untuk membantu dalam membangun sistem yang menggunakan teknologi Computer Vision yang support terhadap Intel Image Processing Library (IPL). OpenCV di desain untuk mengefisienkan komputasi dengan focus terhadap aplikasi yang bersifat real-time.

1.3 MediaPipe

Mediapipe merupakan kerangka kerja sumber terbuka untuk "membangun solusi pembelajaran mesin kelas kata" oleh Google - saat ini dalam tahap alfa. Kini telah menjadi open source selama satu tahun sekarang tetapi kemungkinan telah dikembangkan lebih lama. MediaPipe paket Python bawaan di PyPI. Ini juga menyediakan alat bagi pengguna untuk membangun solusi mereka sendiri.



Sumber : mediapipe.dev

Gambar 1. 3 20 landmarks dari mediapipe

HASIL DAN PEMBAHASAN

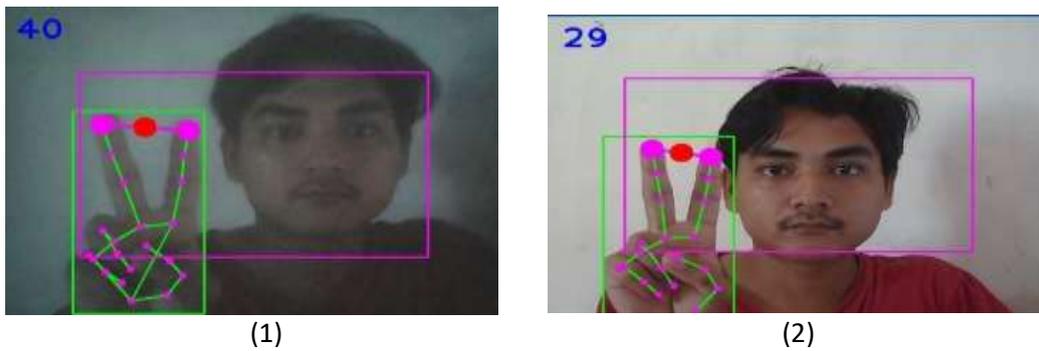
Input dari sistem yang dibangun yaitu berupa citra tangan manusia atau anggota tubuh manusia. Pendeteksian yang dilakukan adalah dengan menggunakan Jari tangan sehingga apabila ada persilangan posisi antara kulit bagian tubuh satu dengan yang lainnya, kemungkinan akan terjadi kesalahan pendeteksian karena adanya nilai warna yang sama dalam ukuran yang besar. penelitian bagian ini sistem deteksi menggunakan algoritma lucas-kanade diuji untuk mengetahui apakah sistem deteksi algoritma lucas-kanade dapat bekerja dengan baik, tahap pengujiannya adalah dengan menggunakan beberapa skenario pengujian yaitu pencahayaan, resolusi kamera dan noise gambar dengan menggunakan webcam beresolusi 2mp dan 6mp.



Gambar 1. 4 pengenalan fungsi objek

2.1 Pengujian Klik

Pengujian dilakukan untuk mengukur tingkat keberhasilan dengan cara melakukan klik kiri, double klik, dan drag pada 10 shortcut yang terdapat pada layar dekstop/komputer. pengklikan dilakukan sebanyak 50 kali masing-masing fungsi klik yang dilakukan di ruangan dengan kondisi penerangan baik dengan menggunakan 2 webcam yang berbeda.



Gambar 1. 2 Gambar (1) merupakan webcam 2mp, gambar (2) merupakan webcam 6mp

Skenario 1: Uji coba dilakukan pada background yang polos tanpa ada benda di sekitar dan kondisi pencahayaan yang kurang. Dengan jarak jari ke kamera 1 meter. Hasil pengujian ditunjukkan oleh Tabel 4.3.

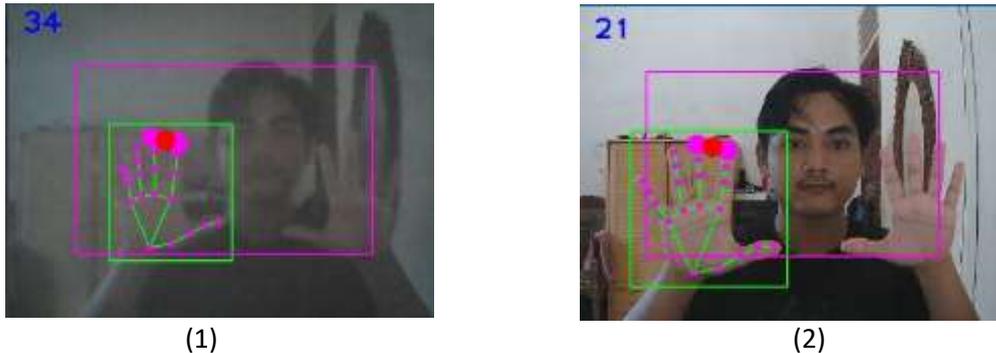
Tabel 1. Hasil presentase dari ketiga fungsi klik

Fungsi klik	Persentase keberhasilan	
	2mp	6mp
Klik kiri	94%	98%
Double klik	90%	94%
Drag	96%	100%
Rata-rata	93,33%	97,33%

Berdasarkan hasil data pengujian pada Tabel 1, terlihat bahwa resolusi webcam yang digunakan berpengaruh terhadap tingkat keberhasilan fungsi klik yang dilakukan. Semakin kecil resolusi atau tingkat ketajaman citra yang dihasilkan oleh webcam digunakan maka akan semakin besar pula tingkat kegagalan yang terjadi. Sebaliknya semakin besar resolusi atau tingkat ketajaman citra yang dihasilkan oleh webcam digunakan maka akan semakin kecil pula tingkat kegagalan. Kesalahan yang terjadi dikarenakan faktor piksel kamera yang di gunakan sehingga area disekitar dapat mempengaruhi kegagalan seperti wajah atau barang yang warna dan bentuk mirip seperti jari.

2.2 Noise dan pencahayaan

Pengujian ini juga sama halnya dilakukan seperti pengujian sebelumnya hanya saja pengujian kali ini terdapat benda-benda yang akan berpotensi menyebabkan adanya kegagalan pada saat pendeteksian jari tangan, cara pengujian yaitu membuka program pada 10 shortcut yang terdapat pada layar sebanyak 50 kali klik yang dilakukan di dalam ruangan dengan adanya benda-benda disekitar, menggunakan 2 webcam yang berbeda. Seperti yang terlihat pada Gambar dibawah ini.



Gambar 1. 2 Gambar (1) webcam 2mp, gambar (2) webcam 6mp

Skenario 2: Uji coba dilakukan pada background yang sudah ada beberapa benda di sekitar dan kondisi pencahayaan yang kurang, normal, dan tinggi . Dengan jarak jari ke kamera 1 meter. Hasil pengujian ditunjukkan oleh Tabel 4.3.

Tabel 2. Hasil presentase dari tingkat pencahayaan

	Pencahayaan		
	tinggi	Sedang	rendah
Webcam 2mp	94%	92%	82%
Webcam 6mp	98%	98%	94%

Berdasarkan hasil data pengujian pada Tabel 2, terlihat dari 50 percobaan yang telah dilakukan ada beberapa kegagalan diantaranya dinyatakan baik sehingga dapat diambil nilai keberhasilan dari webcam 2mp sebesar 89,33% sedangkan dari webcam 6mp sebesar 96,66% , kesalahan dalam hal ini dikarenakan deteksi pada jari tangan terjadi kesalahan pada piksel kamera resolusi rendah mengakibatkan beberapa benda-benda lain yang ada di belakang pengguna dapat terdeteksi sehingga terjadinya kesalahan.

2.3 Faktor faktor yang mempengaruhi

Berdasarkan penelitian yang dilakukan dapat ditemukan faktor-faktor yang mempengaruhi hasil pendeteksian, klik maupun pergerakan. Tingkat kecerahan sangat mempengaruhi tingkat keberhasilan pada pendeteksian *Hand tracking*. Citra yang memiliki tingkat penerangan yang buruk akan menyebabkan kurang maksimalnya citra biner/piksel yang dihasilkan, sehingga perlu diperlukan penyesuaian nilai thresholdingnya. Tingkat kecerahan ini juga mempengaruhi proses pendeteksian jari tangan pengguna, semakin baik tingkat pencahayaan maka akan semakin besar tingkat keberhasilan pendeteksian objek tersebut. Sistem pendeteksian jari tangan yang dibangun hanya dapat mendeteksi jari tangan dengan jarak 1-7 meter. Apabila jari tangan pengguna terlalu jauh, atau terlalu dekat maka tidak akan dapat terdeteksi. Penentuan area letak jari tangan berdasarkan area sekitar yang memiliki benda-benda yang warna dan bentuknya mirip seperti jari tangan berpotensi meningkatkan nilai dari kegagalan dari hasil hand tracking. Jika pergerakan yang dihasilkan pengguna terlalu cepat maka algoritma Lucas-Kanade tidak dapat melakukan penjejakan pada fitur yang telah dideteksi dan akan dilakukan proses penjejakan pada fitur yang baru. Hal ini akan mempengaruhi tingkat stabilitas dalam pergerakan pointer yang dihasilkan.

SIMPULAN

Setelah melakukan beberapa tahap pengujian sistem, maka diperoleh beberapa kesimpulan yang ditemukan pada penelitian ini, yaitu :

1. Algoritma Lucas-Kanade dapat berjalan dengan sangat baik untuk melakukan pelacakan jari tangan.
2. Dalam situasi minim penerangan algoritma lucas-kanade dapat mendeteksi letak jari tangan, dapat di buktikan dengan tingginya nilai rata-rata persentase dari pengujian.
3. Berdasarkan hasil pengujian tingkat keberhasilan melakukan fungsi klik yang dilakukan sebanyak 50 kali percobaan pada setiap fungsi dan dilakukan pada dua webcam yang beresolusi berbeda, diperoleh tingkat keberhasilan pada webcam 6mp sebesar 97,33% sedangkan pada webcam 2mp sebesar 93,33%.
4. Cahaya menjadi salah satu faktor yang mempengaruhi kualitas hasil dari proses yang terjadi didalam sistem. Semakin baik pencahayaannya maka proses komputasi pendeteksian tangan menghasilkan nilai yang baik.
5. Jarak optimal pendeteksian marker didapat yaitu 2-7 meter.
6. Kelanjutan dari penelitian ini akan bisa dikembangkan untuk Aplikasi realtime object tracking seperti wajah dan kedipan mata.

DAFTAR PUSTAKA

- Pamungkas Adi. (2016). Apakah yang dimaksud dengan thresholding??. Retrieved January 05, 2020, from PemogramanMatlab.com website:<https://pemrogramanmatlab.com/pengolahan-citra-digital/segmentasi-citra/thresholding-menggunakan-matlab/>.
- Imaduddin Ahmad. 2013. Pengenalan Jari Tangan Manusia Pada Sistem Kamera Proyektor Sebagai Mouse Pointer. Skripsi. Universitas Islam Negeri Sunan Kalijaga
- Syarif Yuli, dkk. 2013. Perangkat Lunak Pengendali Pointer Menggunakan Pelacakan Mata (Eye Tracking) Dengan Algoritma Lucas Kanade. Jurnal. Universitas Diponegoro Semarang.
- Karisma Hendri. 2011. Implementasi Hand Tracking Pada Kontrol Mouse Pointer Menggunakan Algoritma Pyramidal Lucas-Kanade. Skripsi. Universitas Komputer Indonesia.
- Fatimah Siti. 2017. Interaksi Pada Augmented Reality Satwa Langka Indonesia Dengan Pengenalan Gerakan Jari Tangan. Skripsi. Universitas Sumatra Utara
- Y. Syarif, R. R. Isnanto and K. I. Satoto. 2013. Perangkat Lunak Pengendali Pointer Menggunakan Pelacakan Mata (Eye Tracking) Dengan Algoritma Lucas Kanade. TRANSIENT, vol. 2, no. 3, pp. 714-719
- Kurniawan. Ridho S. 2017. Pengenalan Jari Tangan Sebagai Interaksi pada Augmented Reality Bangunan Heritage Kota Medan Kawasan Matsum dan Kesawan. Skripsi. Universitas Sumatera Utara.
- Fadli feri. Munawir, 2019. Kontrol Mouse Menggunakan Webcam Berdasarkan Deteksi Warna. Jurnal. Fakultas Teknik Universitas Samudra.
- Gita Ivan, dkk. 2012. Implementasi Dan Analisis Hand Tracking Menggunakan Active Contour (Snakes). Jurnal. Universitas Telkom.
- Riskinaswa, L. (2020). Revolusi Industri 4.0. Retrieved January 04, 2020, from kominfo website:<https://aptika.kominfo.go.id/2020/01/revolusi-industri-4-0/>.
- Ichi.pro(2020).mediapipe. Retrieved January 04, 2020. From ichi.pro website :<https://ichi.pro/id/tinjauan-kerangka-ai-ramah-seluler-baru-google-mediapipe-41601805353633>.
- Retrieved January 05, 2020, from OpenCv website:https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html.