

Pemanfaatan *Hand Tracking* untuk Membuat Program *Virtual Painter* sebagai Alternatif Menggambar Digital

Dwi Oktaviyanti^{1*)}; Anan Nugroho¹; Alfa Faridh Suni¹

1. Jurusan Teknik Elektro, Universitas Negeri Semarang, Jawa Tengah 50229, Indonesia

^{*)}Email: dwioktaviyanti@students.unnes.ac.id

Received: 17 November 2021 | Accepted: 2 September 2022 | Published: 28 November 2022

ABSTRACT

Technology will always develop every day. One of the widely used concepts is the Human Machine Interface, for example, the stylus which allows you to draw on a tablet, laptop, or other compatible device. The results of the images that are directly connected to the computer make many people like its usage. However, not many people still use paper for drawing because the stylus is quite expensive and requires a device that is compatible with the stylus. The purpose of this journal is to create a program that allows users to draw digitally without the need for touch screen devices or other tools, just by moving their hands in front of the camera to draw and the results can be directly seen on the computer. The method used is the concept of hand tracking to track the movement of the fingers used for drawing. The results obtained are the program detects 21 keyponits hand landmark users using a 30 fps webcam. These keypoints are analyzed to further determine the current mode (select or draw). The resulting image can be directly seen in the input frame. Program accuracy can reach 100% with medium to high light intensity and with a distance of 50-100 cm.

Keywords: *Hand-tracking, HMI, web camera*

ABSTRAK

Teknologi akan selalu berkembang setiap harinya. Salah satu konsep yang banyak digunakan yaitu Human Machine Interface. Contohnya yaitu stylus yang memungkinkan untuk menggambar di tablet, laptop, atau perangkat kompatibel lainnya. Hasil gambar yang langsung terhubung dengan komputer membuat banyak orang menyukai penggunaannya ini. Namun, tidak banyak orang yang masih menggunakan media kertas untuk menggambar dikarenakan harga stylus yang cukup mahal serta perlu perangkat yang kompatibel dengan stylus tersebut. Tujuan dari jurnal ini adalah pembuatan program yang memungkinkan user untuk menggambar digital tanpa perlu perangkat layar sentuh atau alat lain, hanya dengan menggerakkan tangan di depan kamera untuk menggambar dan hasil gambar dapat langsung terlihat di komputer. Metode yang digunakan yaitu konsep hand tracking untuk malacak pergerakan jari tangan yang digunakan untuk menggambar. Hasil yang diperoleh yaitu program mendeteksi 21 keyponits hand landmark user menggunakan webcam 30 fps. Keypoints tersebut dianalisis untuk selanjutnya menentukan mode yang sedang berjalan (memilih atau menggambar). Gambar yang dihasilkan dapat langsung terlihat dalam frame input. Keakurasian program dapat mencapai 100% dengan intensitas cahaya sedang hingga tinggi serta dengan jarak 50-100 cm.

Kata kunci: *Hand-tracking, HMI, landmark, web camera*

1. PENDAHULUAN

Teknologi akan selalu berkembang setiap harinya. Banyak sekali teknologi yang dikembangkan dapat mempermudah pekerjaan manusia. Dalam mengembangkan teknologi, konsep yang saat ini banyak digunakan yaitu HMI (*Human Machine Interface*). Dengan HMI memungkinkan manusia untuk mengontrol perangkat keras dan mengumpulkan informasi umpan balik [1]. Pengembangan teknologi tidak terlepas dari komputer. Komputer dapat difungsikan seperti manusia. Seperti itulah konsep *Computer vision*. *Computer vision* merupakan salah satu cabang ilmu komputer yang membahas bagaimana sebuah komputer dapat ‘melihat’ seperti halnya manusia [2]. Pada hakikatnya, *computer vision* mencoba meniru cara kerja sistem visual manusia (*Human Vision*) [3].

Salah satu contoh dari teknologi yang berkembang saat ini yaitu *stylus pen*. *Stylus pen* sering digunakan dengan perangkat informasi seluler [4]. Perangkat tersebut seperti tablet, PC, atau perangkat lain yang kompatibel dengan *stylus* tersebut. Biasanya perangkat yang kompatibel adalah perangkat yang memiliki fitur layar sentuh. Oleh karena itu, harga *stylus* ini cukup mahal. Dalam *website* Tokopedia, harga *website* Tokopedia, harga *stylus pen* bervariasi dari puluhan ribu hingga jutaan rupiah [5]. Perangkat yang kompatibel dengannya juga mahal. Tidak semua orang dapat menggunakan *stylus* ini. Berdasarkan permasalahan tersebut maka tujuan dari pembuatan jurnal ini adalah untuk membuat program yang memungkinkan user untuk menggambar digital tanpa perlu perangkat layar sentuh atau alat lain, hanya dengan menggerakkan tangan di depan kamera komputer (*web cam*) untuk menggambar dan hasil gambar dapat langsung terlihat di *frame* kamera *output*.

Penelitian ini merupakan perkembangan bentuk lain yang sederhana dari penelitian jurnal yang berjudul “*VIRTUAL MOUSE USING HAND GESTURE*” oleh Sujithra Sarala dan Venkatesan yang diterbitkan di *International Journal of Modern Trends in Engineering and Research (IJMTER)* [6]. Dalam penelitian tersebut sistem atau program yang dihasilkan berupa *virtual mouse* yang dapat mengontrol kursor mouse menggunakan *webcam* dan sistem tersebut didasarkan pada algoritma *computer vision* yang dapat melakukan semua tugas mouse [6]. Sistem ini dapat berguna dalam presentasi dan mengurangi ruang kerja dengan beberapa fitur yang dapat digunakan seperti memperbesar dan mengecilkan jendela, menutup jendela, dll dengan menggunakan telapak tangan dan beberapa jari [6].

Metode yang digunakan yaitu menerapkan konsep *hand tracking* dalam program untuk malacak pergerakan jari tangan yang digunakan untuk menggambar. *Hand-tracking* dapat dengan mudah ditemukan dan digunakan dalam *library MediaPipe*. *MediaPipe* adalah kerangka kerja (*framework*) untuk membangun *pipelines* untuk melakukan inferensi atas data sensorik arbitrer. Dengan *MediaPipe*, saluran *pipeline* dapat dibangun sebagai grafik komponen modular, termasuk inferensi model, algoritme pemrosesan media, transformasi data, dll. Data sensorik seperti aliran audio dan video memasuki grafik, dan deskripsi yang dirasakan seperti lokalisasi objek dan wajah aliran tengara keluar dari grafik. [7]. *MediaPipe* banyak dimanfaatkan dalam program deteksi. Misalnya seperti deteksi objek, pengenalan wajah, identifikasi gerakan tangan, dll [8]. *MediaPipe* juga dapat digunakan dalam banyak Bahasa pemrograman, seperti C++, Python, JS, dll. Dalam pembuatan program *Virtual Painter* ini menggunakan bahasa pemrograman Python. Python dirancang dengan penekanan pada keterbacaan kode, dan sintaksnya memungkinkan programmer untuk mengekspresikan konsep mereka dalam baris kode lebih sedikit [9]. Selain *library MediaPipe*, dalam program juga menggunakan *library OpenCV*. *OpenCV (Open-Source Computer vision Library)* merupakan salah satu *library* yang dibuat oleh Intel, yang mana *library* ini digunakan untuk pemrosesan gambar secara *real-time* yang dinamis, dan sekarang didukung oleh Willow Garage dan Itseez [10].

2. METODE/PERANCANGAN PERCOBAAN

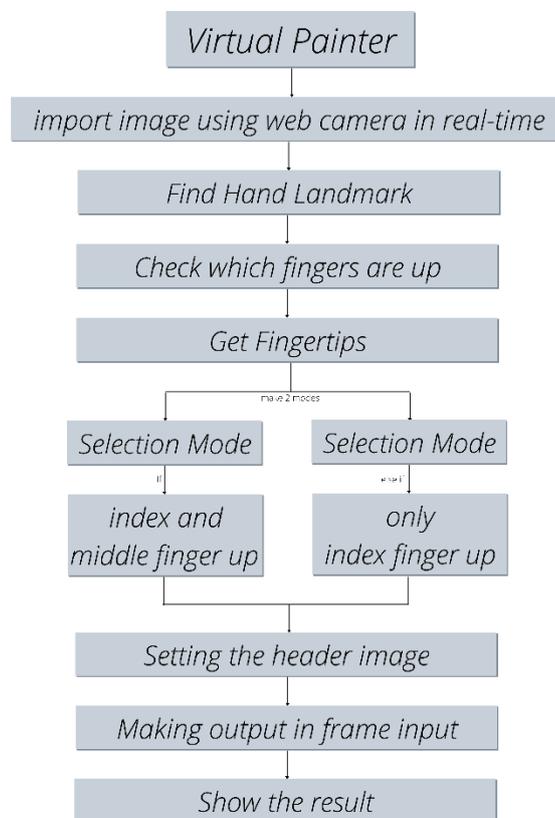
2.1. Alat dan Bahan

Alat dan bahan yang digunakan dalam percobaan ini yaitu:

- Laptop yaitu Laptop Asus X453MA Processor Intel(R) Celeron(R) CPU N2840 @ 2.16GHz, 2159 Mhz, 2 Core(s), 2 Logical Processor(s), dan Installed Physical Memory (RAM) 8.00 GB.
- Software PyCharm 2021.2.1 (Community Edition)
- File header yang digunakan untuk memilih warna cat dan mode penghapus.
- Webcam (dalam percobaan ini penulis menggunakan software DriodCam yang terinstall baik di laptop maupun smart phone) dengan 30 frames/second, dan resolusi 640x480.

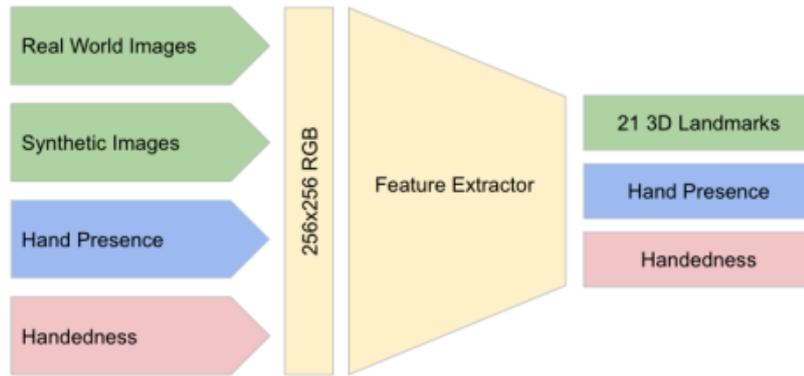
2.2. Skenario Percobaan

Rancangan skenario dalam percobaan ini dapat dilihat pada gambar 1 berikut.



Gambar 1. Alur Skenario Program Virtual Painter

MediaPipe Hands adalah solusi pelacakan tangan dan jari dengan ketelitian tinggi. Ini menggunakan *machine learning* (ML) untuk menyimpulkan 21 *landmark* 3D tangan hanya dari satu bingkai. MediaPipe Hands terdiri dari beberapa model yaitu Model *Palm Detection* yang beroperasi pada gambar penuh dan mengembalikan kotak pembatas tangan yang berorientasi. Serta Model *Hand Landmarks* yang beroperasi pada area gambar yang dipangkas yang ditentukan oleh detektor telapak tangan dan mengembalikan titik kunci tangan 3D dengan fidelitas tinggi [11]. Model *hand landmarks* ini memiliki tiga output yaitu 21 *landmark* tangan yang terdiri dari x, y, dan kedalaman relative (*depth*); bendera tangan yang menunjukkan kemungkinan kehadiran tangan pada gambar masukan; serta klasifikasi biner dari tangan mana yang sedang digunakan, mis. tangan kiri atau kanan [12].



Gambar 2. Arsitektur *hand landmark*



Gambar 3. Point kunci *hand landmark*

Video pada dasarnya merupakan tayangan slide gambar yang ditampilkan pada frekuensi yang cukup, sehingga aliran video perlu dikompres dan *frame rate* yang umum digunakan dalam video yaitu 23-30 fps [13]. Biasanya *frame rate* yang tinggi untuk meningkatkan kualitas yang dirasakan dari konten video tertentu [14]. Dalam program ini, *frame-rate* didapatkan menggunakan *Time Module* pada Python dengan menggunakan metode *time.time()*. Perhitungan yang dilakukan dalam program terlihat pada persamaan berikut [15].

$$cTime = time.time() \tag{1}$$

$$fps = 1 / (cTime - pTime) \tag{2}$$

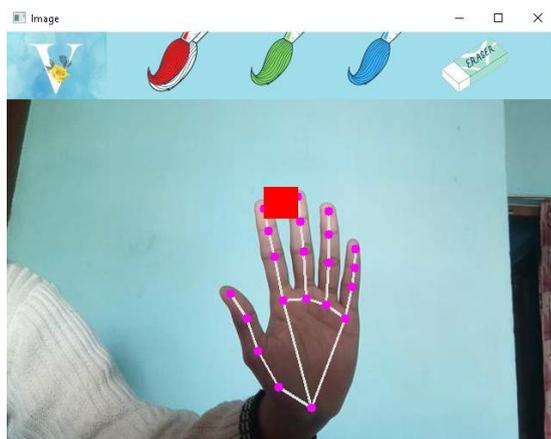
$$pTime = cTime \tag{3}$$

3. HASIL DAN PEMBAHASAN

Hasil yang diperoleh yaitu program memiliki dua mode, mode memilih dan mode menggambar. Mode memilih ketika jari telunjuk dan tengah berdiri, digunakan untuk memilih warna atau penghapus. Mode menggambar ketika jari telunjuk saja yang berdiri. Hasil gambar dapat terlihat secara real-time di frame kamera output.

Dalam meng-*import image*, menggunakan method *VideoCapture()* dari library OpenCV. Dengan method tersebut memungkinkan untuk program dapat mengakses kamera *webcam* yang digunakan.

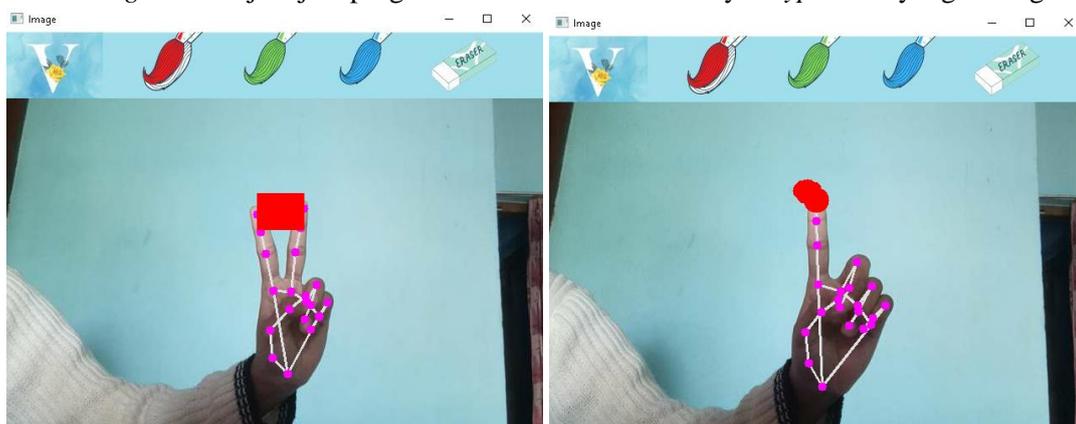
Hand landmark dapat terdeteksi menggunakan *class findHand* yang dibuat menggunakan *MediaPipe Hands*. Berdasarkan gambar 3, hasil dari *hand landmark* berupa titik-titik kunci yang selanjutnya, *keypoints* tersebut buat *conditional statement* untuk membuat *selecting mode* dan *drawing mode*.



Gambar 4. Hand landmark terdeteksi dalam hasil program

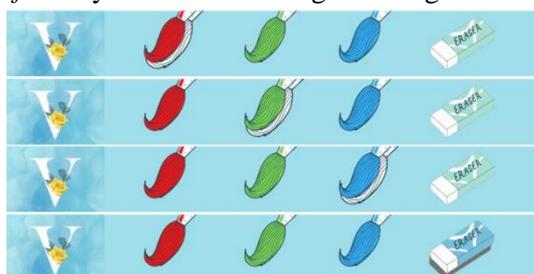
Selecting mode dapat terjadi jika program mendeteksi bahwa hanya *keypoint* 8 dan 12 yang sedang berdiri, ditandai keduanya membuat suatu *rectangle*.

Drawing mode terjadi jika program mendeteksi bahwa hanya *keypoints* 8 yang sedang berdiri.



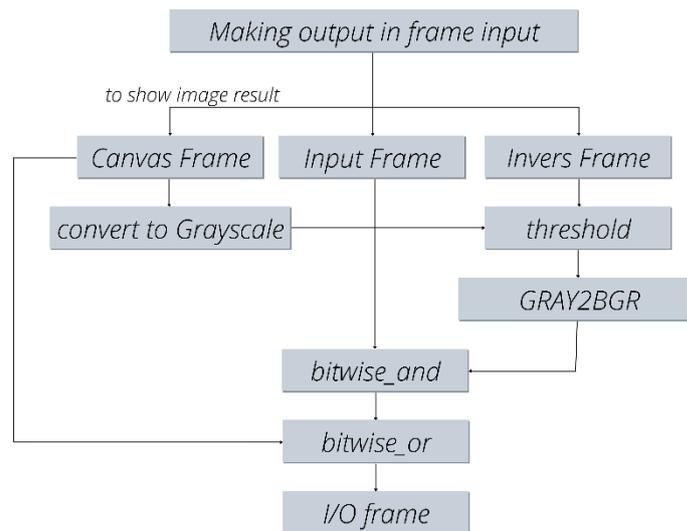
Gambar 5. *Selecting Mode* (kiri) dan *Drawing Mode* (kanan)

Ukuran *header* program diatur berdasarkan ukuran *frame* yang digunakan dalam *webcam*. Dalam program ini, ukuran *frame* yaitu 640x480 dengan hasil gambar berupa BGR.

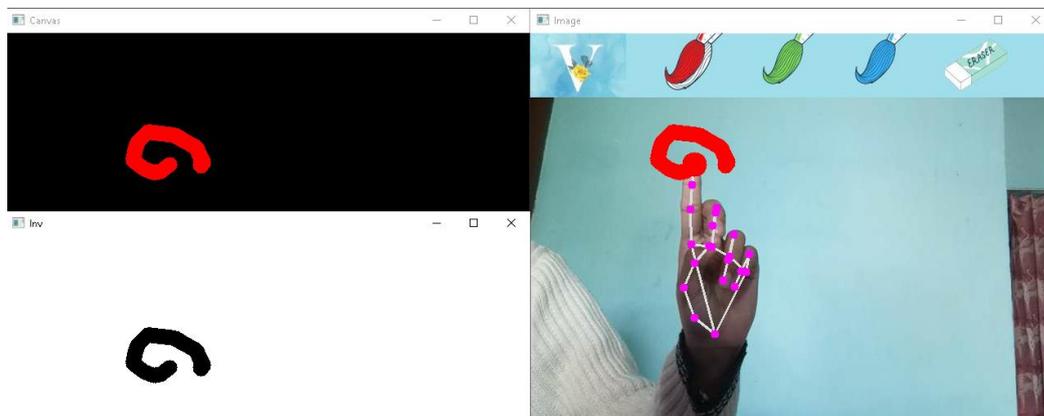


Gambar 6. Header

Hasil gambar yang telah dibuat user dapat tertampil dalam *frame* kamera *input*. Hal tersebut dikarenakan sebelumnya program membuat dua *frame* baru yaitu *frame canvas* dan *frame invers*. Proses dalam menampilkan *output* gambar dapat dilihat pada gambar blok diagram berikut.

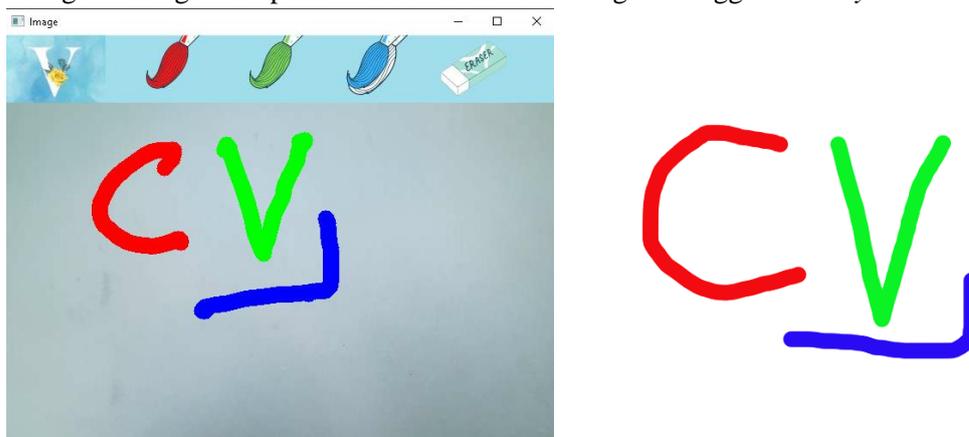


Gambar 7. Blok Diagram untuk membuat I/O frame



Gambar 8. Frame canvas, Frame Invers, dan I/O frame

Perbandingan hasil gambar pada *Virtual Painter* dan dengan menggunakan *stylus*.



Gambar 9. Output Gambar Program *Virtual Painter* (kiri) dan dengan *stylus* (kanan)

Berdasarkan gambar 9, salah satu keuntungan menggambar dengan menggunakan program *Virtual Painter* yaitu latar belakang gambar dapat berupa hasil tangkapan layar kamera secara

realtime. Sedangkan dengan *sylus*, latar belakang perlu diatur secara terpisah agar sesuai keinginan *user*.

Program diuji coba untuk mengukur fungsionalitas fitur-fitur dalam program. Mulai dari mendeteksi *hand landmark user* hingga *user* dapat menggambar dengan lancar. Dalam pengujian ini, parameter yang digunakan yaitu intensitas cahaya dan jarak. Terdapat tiga variasi intensitas cahaya yaitu tinggi, sedang, dan rendah dengan masing-masing variasi tersebut dilakukan percobaan untuk jarak 50 cm, 100 cm, dan 150 cm. Untuk mengukur tingkat akurasi program menggunakan persamaan di bawah.

$$\text{Tingkat Akurasi (\%)} = \frac{\text{Jumlah Terdeteksi}}{\text{Jumlah Percobaan}} \times 100\% \tag{4}$$

Percobaan masing-masing variasi dilakukan sebanyak 20 kali. Sehingga, hasil yang didapatkan terlihat pada tabel 1 berikut.

Tabel 1. Hasil Akurasi Program untuk Mendeteksi *Hand Landmark User*

Cahaya	Jarak (cm)	Jumlah Terdeteksi	Jumlah Percobaan	Akurasi	Rata-rata
Tinggi	50	20	20	100	96.666667
Tinggi	100	20	20	100	
Tinggi	150	18	20	90	
Sedang	50	20	20	100	93.333333
Sedang	100	18	20	90	
Sedang	150	18	20	90	
Rendah	50	12	20	60	33.333333
Rendah	100	6	20	30	
Rendah	150	2	20	10	
Rata-rata Total				74.44444444	

Berdasarkan tabel 1, tingkat akurasi tertinggi yaitu 100% dengan intensitas cahaya tinggi pada jarak 50 cm dan 100 cm serta intensitas cahaya sedang pada jarak 50 cm. Pada saat cahaya berintensitas rendah, tingkat akurasi menjadi kecil. Dengan demikian, program *Virtual Painter* kurang cocok digunakan saat intensitas cahaya rendah karena program tidak bisa mendeteksi *hand landmark user*.

4. KESIMPULAN DAN SARAN

Berdasarkan gambar 9, program *Virtual Painter* dapat dijadikan solusi untuk menggambar secara digital tanpa perlu alat lain seperti *stylus* ataupun perangkat yang memiliki spesifikasi khusus seperti harus bisa layar sentuh. Program ini memanfaatkan konsep *hand tracking* dengan menggunakan *library* MediaPipe untuk melacak pergerakan tangan *user*. Hasil dari pelacakan berupa *hand landmark* yang memiliki 21 *keypoints* dimana dalam program ini. Jika *keypoint* 8 dan 12 yang berdiri, maka program berada pada *selecting mode*. Jika program mendeteksi bahwa hanya *keypoint* yang berdiri, maka program berada pada *drawing mode*. Keuntungan terbesar menggambar melalui program ini dibandingkan menggunakan *stylus* selain karena tidak perlu mengeluarkan biaya yang banyak, latar belakang gambar yang dihasilkan dapat disesuaikan secara *realtime* melalui *webcam*. Sedangkan jika dengan menggunakan *stylus*, latar belakang perlu diatur atau ditambahkan secara terpisah dari hasil gambar. Serta berdasarkan tabel 1, keakurasian program dapat mencapai 100% dengan intensitas cahaya sedang hingga tinggi dengan jarak 50-100 cm.

Disarankan dalam penelitian ini yaitu :

- a. Penelitian perlu dilanjutkan agar program dapat menyimpan hasil gambar yang telah dibuat.
- b. Peneliti n juga dapat dikembangkan agar *user* dapat mengatur ketebalan kuas yang diinginkan.

DAFTAR PUSTAKA

- [1] S. Lim, D. Son, J. Kim, Y. B. Lee, J.-K. Song, S. Choi, D. J. Lee, J. H. Kim, M. Lee, T. Hyeon dan D.-H. Kim, "Transparent and Stretchable Interactive Human Machine Interface Based on Patterned Graphene Heterostructures," *Advanced Functional Materials*, vol. 25, no. 3, pp. 375-383, 2015.
- [2] D. A. Wahyudi dan I. H. Kartowisastro, "MENGHITUNG KECEPATAN MENGGUNAKAN COMPUTER VISION," *Jurnal Teknik Komputer*, vol. 19, no. 2, pp. 89 - 101, 2011.
- [3] K. D. Irianto, "PENDETEKSI GERAK BERBASIS KAMERA MENGGUNAKAN OPENCV PADA RUANGAN," *KomuniTi*, vol. 2, no. 1, pp. 52-59, 2010.
- [4] E. Park, A. P. D. Pobil dan S. J. Kwon, "Usability of the Stylus Pen in Mobile Electronic Documentation," *Electronics*, vol. 4, no. 4, pp. 922-932, 2015.
- [5] Tokopedia, Tokopedia, [Online]. Available: <https://www.tokopedia.com/find/stylus-pen>. [Diakses 6 11 2021].
- [6] S. Sujithra dan Venkatesan, "VIRTUAL MOUSE USING HAND GESTURE," *International Journal of Modern Trends in Engineering and Research (IJMTER)*, vol. 4, no. 3, pp. 5-9, 2017.
- [7] C. Lugaresi, . J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, L. Juhyun, W.-T. Chang, W. Hua, M. Georg dan M. Grundmann, "MediaPipe: A Framework for Building Perception Pipelines," *arXiv preprint arXiv:1906.08172*, pp. 1-9, 2019.
- [8] MediaPipe, "Live ML anywhere," MediaPipe, [Online]. Available: <https://mediapipe.dev/>. [Diakses 6 11 2021].
- [9] F. L. Ahmad, A. Nugroho dan A. F. Suni, "Deteksi Pemakai Masker Menggunakan Metode Haar Cascade Sebagai Pencegahan COVID 19," *Edu ElektriKa Journal*, vol. 10, no. 1, pp. 13-18, 2021.
- [10] F. Indaryanto, A. Nugroho dan A. F. Suni, "Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19," *Edu Komputika Journal*, vol. 8, no. 1, pp. 31-38, 2021.
- [11] "MediaPipe Hands," MediaPipe, 2020. [Online]. Available: <https://google.github.io/mediapipe/solutions/hands.html>. [Diakses 9 October 2021].
- [12] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang dan M. Grundmann, "MediaPipe Hands: On-device Real-time Hand Tracking," *arXiv:2006.10214v1 [cs.CV]*, pp. 1-5, 2020.
- [13] E. Chromy, J. Suran, M. Kovacik dan M. Kavac, "Usage of Erlang Formula in IP Networks," *Communications and Network*, vol. 3, pp. 161-167, 2011.
- [14] A. Mackin, F. Zhang dan D. R. Bull, "A Study of High Frame Rate Video Formats," *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1499-1512, 2019.
- [15] M. Hassan, "AI Virtual Painter," *CV Zone*, [Online]. Available: <https://www.computervision.zone/courses/ai-virtual-painter/>. [Diakses 9 October 2021].