

# Rancang Bangun *Ai Virtual Mouse* Berbasis Pengolahan Citra

## *Design Of Ai Virtual Mouse Based Image Processing*

1<sup>st</sup> Aldy Andreansyah Himawan

Fakultas Ilmu Terapan

Universitas Telkom

Bandung, Indonesia

aldyhimawan@student.telkomuniversity.ac.id

2<sup>nd</sup> Suci Aulia

Fakultas Ilmu Terapan

Universitas Telkom

Bandung, Indonesia

suciaulia@telkomuniversity.ac.id

3<sup>rd</sup> Muhammad Iqbal

Fakultas Ilmu Terapan

Universitas Telkom

Bandung, Indonesia

miqbal@telkomuniversity.ac.id

**Abstrak**—Teknologi akan terus berkembang seiring berjalannya waktu, oleh karena itu masyarakat harus bisa beradaptasi akan hal tersebut. Salah satunya komputer, tidak banyak orang yang bisa menggunakan alat tersebut. Ini menyebabkan kurangnya masyarakat dalam pengetahuan tentang teknologi, oleh sebab itu Human Computer Interaction (HCI) tidak lepas dari komputer. HCI ilmu yang mempelajari suatu hubungan agar memudahkan manusia untuk menggunakan komputer. Salah satu teknologi komunikasi yang dapat berinteraksi secara langsung yaitu virtual. Sehingga untuk mengatasi masalah tersebut dilakukan upaya interaksi dengan komputer agar masyarakat dapat memahaminya. Pada proyek akhir ini, telah dirancang bangun virtual berbasis pengolahan citra menggunakan bahasa pemrograman python. Rancangan ini yang nantinya akan berinteraksi antar manusia dengan komputer secara langsung, dengan menggunakan kamera atau webcam. Untuk melakukan hal tersebut dibutuhkan objek penggerak dan warna sebagai media penandanya. Untuk mendeteksi warna melalui webcam atau kamera yang ditangkap secara real-time. Pada proyek akhir ini, telah dilakukan pengujian klik kursor dengan kondisi indoor dan outdoor menggunakan warna merah, hijau, biru, kuning, dan merah muda memiliki tingkat akurasi 100% dengan jarak terbaik antara kamera laptop atau smartphone dengan warna citra adalah 25 cm, dan pengujian drag kursor dengan kondisi dan warna citra yang sama memiliki tingkat akurasi 85%.

**Kata kunci** — *Virtual, webcam, pengolahan citra*

*Abstract*—Technology will continue to develop overtime, therefore people must be able to adapt. One of them is a computer, not many people can use the tool. This leads to a lack of society in knowledge of technology, therefore Human Computer Interaction (HCI) is inseparable from computers. HCI a science that studies relationships to make it easier for humans to use computers. One of the communication technologies that can interact directly is virtual. To overcome this problem, efforts are made to interact

*with computers so the public can understand it. In this final project, a virtual based on image processing has been designed using the python programming language. This design will later interact between humans and the computer directly, using a camera or webcam. A driving object and color are needed as the marking medium. To detect colors through a webcam captured in realtime. This final project, cursor click testing has been carried out with indoor and outdoor conditions using red, green, blue, yellow, and pink colors has 100% accuracy with the best distance between laptop cameras or smartphones with image color is 25cm, and drag cursor testing with the same image condition and color has an accuracy 85%.*

**Keyword** — *Virtual, webcam, image processing*

### I. PENDAHULUAN

*AI (Artificial Intelligence)* ialah salah satu bidang ilmu komputer yang terkait dengan kecerdasan manusia seperti pembelajaran, pemecahan masalah, maupun pengenalan pola[1]. Oleh karena itu *AI* dapat di implementasikan untuk *AR* dan *VR*. *VR (Virtual Reality)* suatu teknologi yang dapat membawa seseorang ke dalam dunia berbeda melalui perangkat khusus yang telah disesuaikan. Sedangkan *AR (Aumented Reality)* suatu Teknologi yang dapat mewujudkan berbagai objek, lokasi dan imajinasi buatan ke dalam dunia nyata secara langsung. *AR* dan *VR* untuk pengoperasiannya menggunakan sebuah alat bernama komputer.

Komputer sudah menjadi alat wajib bagi mereka yang suka dengan alat-alat elektronik. Komputer memiliki beberapa *hardware* diantaranya layar, *keyboard*, *mouse*, dsb. Komputer memiliki dua versi, yakni versi lama dan versi baru, sama halnya dengan *hardware*, sebagai contoh *mouse*. *Mouse* salah satu *hardware* penting untuk menjalankan

sebuah komputer. *mouse* memiliki dua jenis, yaitu *mouse reality* merupakan *mouse* yang memiliki fisik contohnya *mouse wireless* dan *mouse cable*, dan *mouse virtual* merupakan *mouse* yang tidak memiliki fisik, sehingga dengan adanya *mouse virtual* dapat mengakses komputer tanpa berinteraksi secara langsung, sehingga dapat menjadi solusi pada kondisi pandemi saat ini.

Selain itu, komputer dapat memudahkan seseorang dalam bekerja maupun bersekolah. Untuk menyempurnakan *mouse virtual*, *image processing* akan mengelolah gambar ke dalam bentuk digital. Oleh karena itu, *mouse* berbasis pengolahan citra (*image processing*) yang dapat disebut *AI virtual mouse* menjadi solusi untuk masalah ini. *Virtual mouse* merupakan perangkat lunak komputer yang bisa melakukan interaksi secara langsung pada manusia dan komputer dengan menggunakan perantara *webcam* atau kamera[2]. *Virtual mouse* bertujuan untuk mengontrol suatu *interface* dari jarak jauh tanpa menggunakan alat elektronik, melainkan menggunakan tangan.

Pada penelitian sebelumnya pernah dilakukan, dengan judul "Virtual Mouse" yaitu menggunakan *blob detector* sebagai pendeteksinya[3]. Pada proyek ini, dilakukan pendeteksian menggunakan RGB sebagai penandanya disetiap jarinya untuk menggerakkan kursor di desktop secara *realtime*.

Pada proyek ini, dilakukan rancangan sistem pendeteksi tangan atau jari digerakkan di depan kamera untuk menggerakkan kursor pada komputer sehingga dapat memindahkan ataupun menekan aplikasi atau *file*. Dengan adanya teknologi ini memudahkan pengguna dalam bekerja, sehingga tidak perlu menyentuh elektronik untuk mengakses komputer ditempat umum maupun dikantor sekalipun.

## II. KAJIAN TEORI

### A. Artificial Intelligence

Kecerdasan Buatan "Artificial Intelligence" pertama kali dikemukakan pada tahun 1956 di konferensi Darthmouth. Selama bertahun-tahun para filsuf berusaha mempelajari kecerdasan buatan yang dimiliki manusia. Dari pemikiran tersebut lahirlah AI sebagai cabang ilmu yang berusaha mempelajari dan meniru kecerdasan manusia. Sejak saat itu para peneliti mulai memikirkan perkembangan AI sehingga teori-teori dan prinsip-prinsipnya berkembang terus hingga sekarang[1].

Kecerdasan buatan berasal dari Bahasa inggris "Artificial Intelligence" atau disingkat AI. *Intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud disini merujuk pada mesin yang mampu berfikir, menimbang tindakan yang akan diambil, dan mampu mengambil keputusan seperti yang dilakukan oleh manusia[1].

Pada proyek akhir ini AI suatu teknik mengenali suatu objek, sehingga dapat berpikir seperti manusia, dengan mengklasifikasi kursornya untuk bergerak kearah kiri dan kanan serta menekan dengan cara diberi pola dari citra yang berwarna pada jari, lalu dimasukkan ke *Machine Learning* yaitu komputer.

### B. Pycharm

Pycharm merupakan perangkat lunak IDE (*integrated development environmental*) untuk sistem operasi windows sehingga membuatnya lebih mudah untuk menggunakan dan membuat program dengan bahasa pemrograman *python*[4].



GAMBAR 2.1  
LOGO PYCHARM (SOURCE: PNGAAA.COM)

Pycharm merupakan IDE milik perusahaan JetBrains, pemilik IntelliJ IDEA. Pycharm memudahkan pengguna untuk belajar *deep learning* maupun *machine learning*[4].

### C. OpenCV

*OpenCV* adalah sebuah *software* pustaka untuk *image processing*. *OpenCV* bisa digunakan di *Windows*, *Linux*, *Mac OS X*, *Android*, *iOS*. Bahasa yang digunakan C++, *Python*, *Julia*, *Javascript*. *OpenCV* dapat medeteksi identitas (jarak, pengenalan, warna, dll) objek yang terlihat di kamera[5].



GAMBAR 2.2  
LOGO OPENCV (SOURCE: PNGAAA.COM)

### D. Image Processing

*Image processing* atau pengolahan citra merupakan suatu metode atau teknik yang dapat digunakan untuk memproses citra atau gambar dengan jalan memanipulasinya menjadi suatu data gambar yang diisikan untuk mendapatkan suatu informasi tertentu mengenai objek yang sedang diamati[6]. Warna adalah kesan yang diperoleh mata dari cahaya yang dipantulkan oleh benda-benda yang dikenalnya[7]. Warna yang diperoleh

mata dari sebuah objek ditentukan oleh warna sinar yang dipantulkan oleh objek tersebut. Warna-warna yang dapat ditangkap oleh mata manusia merupakan kombinasi cahaya dengan panjang berbeda.

Warna citra RGB (*Red, Green, Blue*) adalah dengan mengalikan jumlah pada masing-masing komponennya, jumlah dari tiap komponennya,  $R=255$  (8 bit)  $G=255$ (8 bit) dan  $B=255$ (8 bit) sehingga sering kali disebut citra dengan intensitas 24 bit. Dengan nilai minimum 0 dan nilai maksimum 255. Banyaknya kemungkinan minimum dan nilai maksimumnya bergantung pada jumlah bit yang digunakan[7].

HSV (*Hue, Saturation, Value*) ruang warna adalah mendekati ruang warna RGB dimana manusia menggambarkan warna sensasi dan persepsi warna[8]. Hue adalah warna yang dominan diamati oleh manusia, saturasi adalah jumlah cahaya putih bermacam-macam dengan rona, nilai adalah kecerahan / Intensitas. Singkatnya, hue mengacu pada warna, Saturasi mengacu pada bayangan dan Nilai mengacu pada nada. Ruang warna HSV dapat dilihat sebagai geometris silinder, di mana dimensi sudut mewakili Hue(H), mulai dari merah primer pada  $0^\circ$ , dan pindah ke hijau primer pada  $120^\circ$  dan biru primer pada  $240^\circ$ , dan akhirnya kembali menjadi merah pada  $360^\circ$ . Jarak dari sumbu pusat HSV silinder sesuai dengan Saturasi (S). Nilai saturasi bergerak menuju tepi luar berarti bahwa warna-warni nilai maksimum untuk warna yang ditentukan oleh rona. Sumbu vertikal tengah ruang warna HSV adalah Nilai (V), mulai dari hitam di bagian bawah dengan lightness atau nilai 0, hingga putih di bagian atas dengan ringan atau nilai 1[8].

### 2.5 Python

*Python* salah satu Bahasa pemrograman tingkat tinggi yang dapat eksekusi instruksi multi secara langsung serta dapat membaca syntax secara dinamis. *Python* terdapat sistem operasi seperti *Windows, Linux, Android, Mac OS*, dll[9].



GAMBAR 2. 3  
LOGO PYTHON (SOURCE : PNGAAA.COM)

Pemrograman ini dapat menjalankan fungsi kompleks dengan mudah, dan menggunakan sedikit kode. *Python* sering diterapkan dalam pembuatan apk *smartphone, software, IOT, games*, dll . *Python* dibuat oleh Guido van Rossum pada tahun 1991.

## III. METODE

### A. Sistem Perancangan *Virtual Mouse*

Adapun sistem rancangan virtual mouse dapat dilihat pada gambarsebagai berikut.



GAMBAR 3. 1  
BLOK DIAGRAM PERANCANGAN *VIRTUAL MOUSE*

#### 1. *Input*

Pada bagian ini yang akan digunakan sebagai pendeteksinya yaitu warna dan tangan sebagai media penggerakannya.

#### 2. Filter degradasi warna

Mencari nilai batas bawah dan batas atas (*lowerbound* dan *upperbound*) untuk setiap warna, yaitu warna merah, hijau, biru, kuning, dan merah muda.

#### 3. Deteksi

Selanjutnya *webcam* yang berfungsi sebagai pendeteksi warna. Lalu warna yang dideteksi akan masuk ke pengolahan citra.

#### 4. Pengolahan Citra

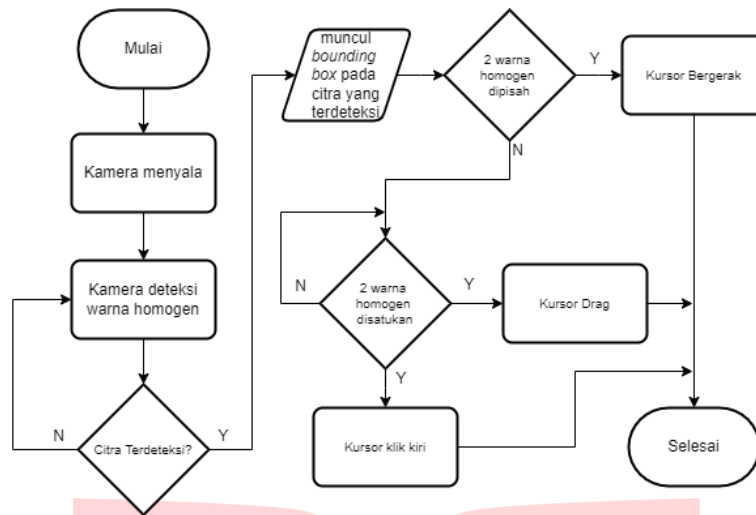
Pada bagian ini dilakukan pemrograman agar mouse bisa bergerak melalui *webcam*. Terdapat laptop atau *PC* sebagai alat, *pycharm* sebagai *software* untuk dilakukannya program, *python* sebagai bahasanya, dan *OpenCV* sebagai *library*.

#### 5. *Output*

Keluaran yang dihasilkan tampilan *mouse* dapat bergerak menggunakan media tangan dan warna sebagai citranya secara *realtime*.

### B. Diagram Alur *Virtual Mouse*

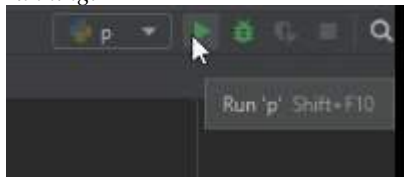
Adapun diagram alir pada *virtual mouse* ini sebagai berikut.



GAMBAR 3.2  
FLOWCHART VIRTUAL MOUSE

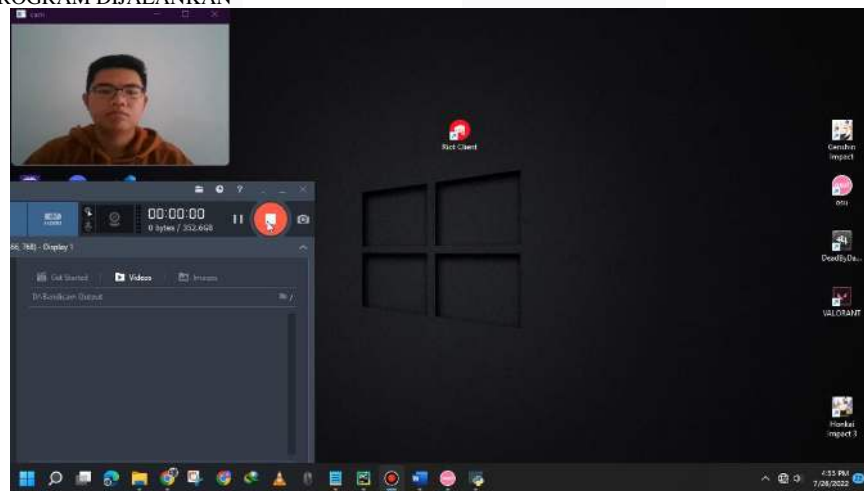
Akan dijelaskan tahapan dan alur proses pada *flowchart virtual mouse* diatas, berikut penjelasannya.

1. Mulai  
Pada bagian ini program memulai *running*.



GAMBAR 3.3  
PROGRAM DIJALANKAN

2. Kamera menyala  
Saat program dijalankan untuk membuka kamera laptop atau *smartphone* (tergantung kamera yang dipakai) biasanya membutuhkan waktu satu detik, apabila ada beberapa program seperti *game, google, video*, dsb yang sedang berjalan, maka membutuhkan waktu empat detik.



GAMBAR 3.4  
KAMERA SEDANG *RUNNING*

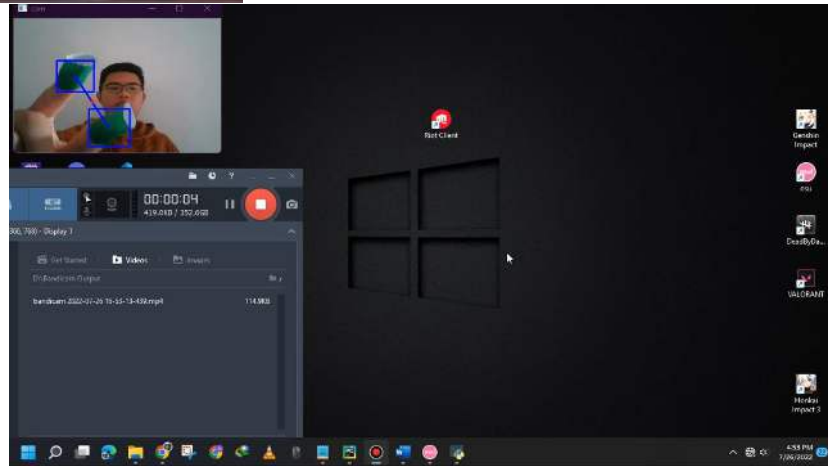
3. Kamera deteksi warna homogen  
Selanjutnya kamera siap mendeteksi warna citra yang sama, apabila benar warna citra yang dideteksi, maka program mengeluarkan *bounding box* atau kotak sebagai penanda batasan deteksi yang ditangkap oleh kamera. Apabila nilai warna HSV

berbeda dengan warna yang akan di deteksi, maka *bounding box* keluar pada warna yang sesuai dengan nilai HSV dan warna yang tidak sesuai, maka kamera tidak mendeteksi warna citra.



GAMBAR 3.5  
CITRA TERDETEKSI DAN  
MENGELUARKAN *BOUNDING BOX*

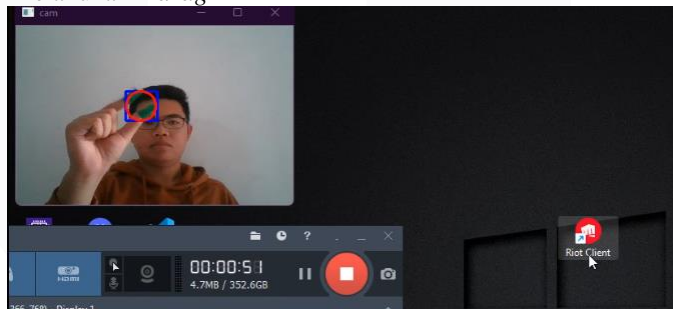
4. Dua warna homogen terpisah  
Bila terdapat dua warna citra yang sama terpisah maka kursor mengalami aksi yaitu kursor dapat di gerakan dengan patokan *red dot* pada kedua warna citra tersebut.



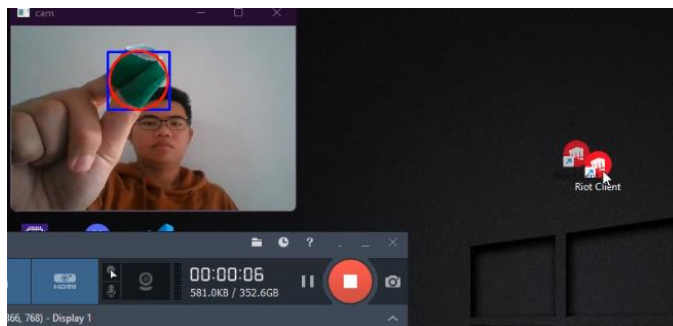
GAMBAR 3.6  
TERDAPAT *RED DOT* DIANTARA 2 CITRA YANG SAMA

5. Dua warna homogen disatukan  
Bila kedua warna citra disatukan maka kursor mengalami 3 aksi yaitu kursor dapat melakukan klik, kursor dapat melakukan *drag*

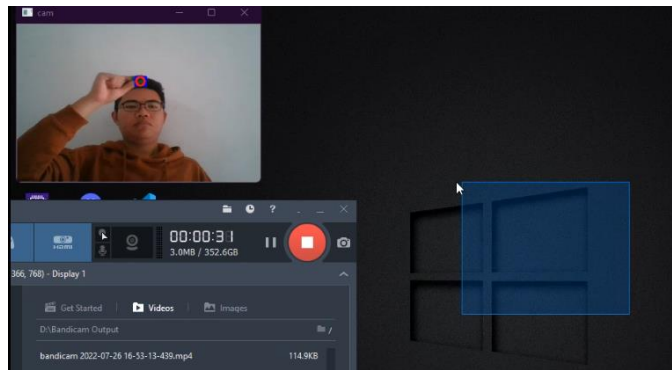
dengan syarat harus ada objek, dan kursor dapat melakukan *selection* dengan syarat harus tidak ada objek dan *mouse* tetap digerakkan.



GAMBAR 3.7  
AKSI *LEFT CLICK* KURSUS

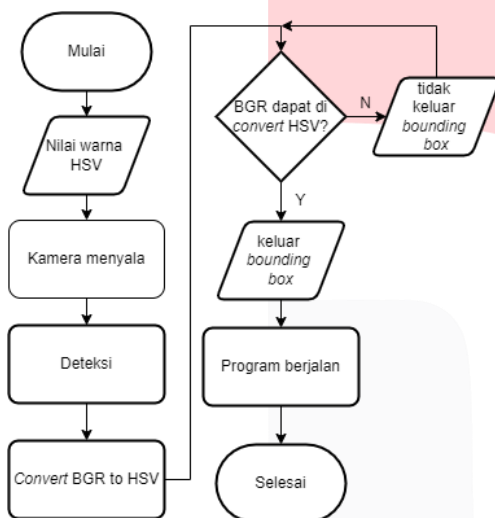


GAMBAR 3.8  
AKSI *DRAG* KURSUS



GAMBAR 3. 9  
AKSI SELECTION KURSOR

C. Diagram Alur *Software* Pycharm  
 Pada bagian ini dilakukan *breakdown flowchart virtual mouse* sebagai berikut.

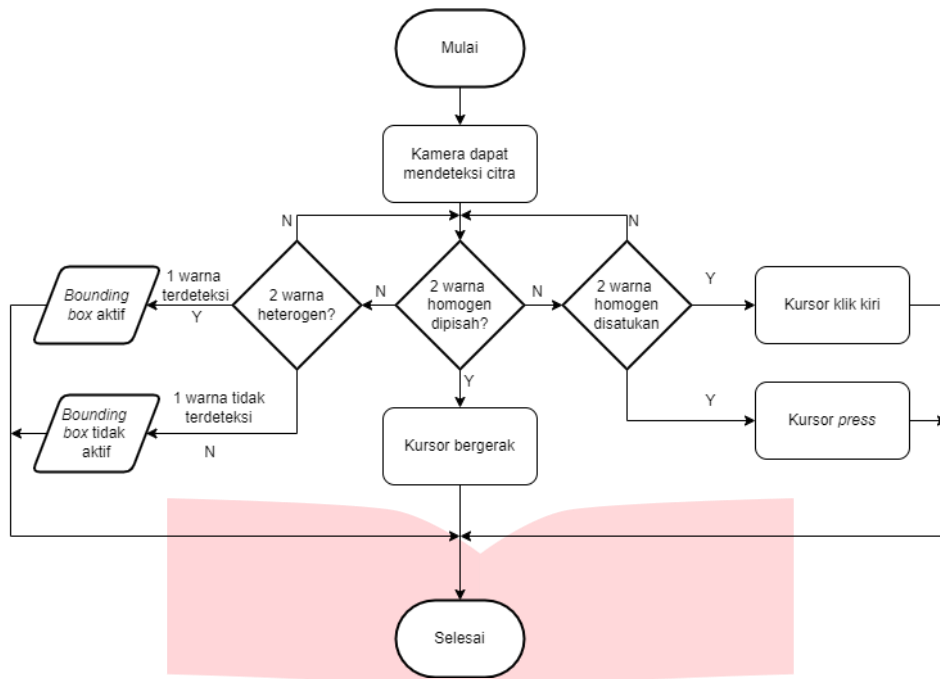


GAMBAR 3. 10  
FLOWCHART PROSES KAMERA MENDETEKSI WARNA

Akan dijelaskan tahapan dan alur proses pada *flowchart* proses kamera mendeteksi warna diatas, berikut penjelasannya.

1. *Input* nilai warna HSV sebelum program dijalankan atau *running*, masukkan terlebih dahulu

- nilai warna yang hendak di deteksi sesuai dengan nilai ambang batas warna citra HSV.
2. Kamera menyala  
Lalu dilakukan *running* untuk membuka kamera.
3. Proses deteksi  
Pada saat kamera terbuka, kamera siap untuk mendeteksi warna citra. Sesuai dengan program yang telah dibuat.
4. *Convert* BGR to HSV  
Pada bagian ini *opencv* membaca warna RGB menjadi BGR, warna BGR yang telah ditangkap oleh kamera diubah dalam bentuk nilai warna HSV sesuai dengan *upper* dan *lower* atau nilai ambang batas dari HSV tersebut. Apabila warna BGR dapat di convert menjadi nilai warna HSV, maka keluaran yang dihasilkan berupa *bounding box*, begitu juga sebaliknya.
5. Program berjalan  
Pada bagian ini masuk tahap pemilihan fitur *mouse* yang akan dijalankan.

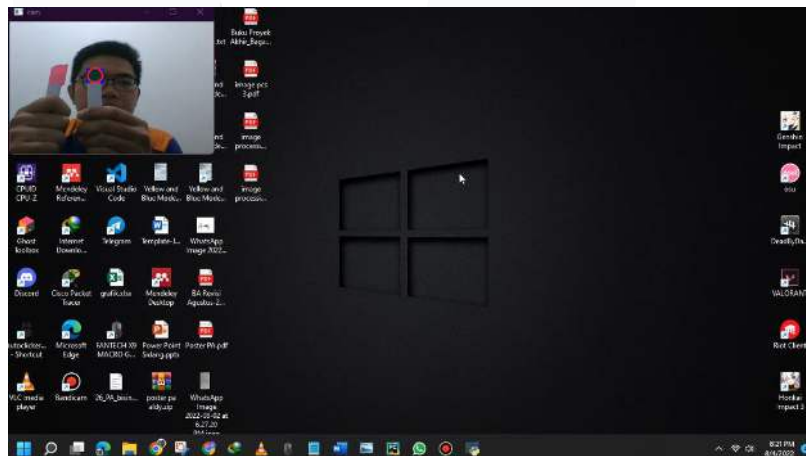


GAMBAR 3. 11  
FLOWCHART PROSES WARNA TERDETEKSI

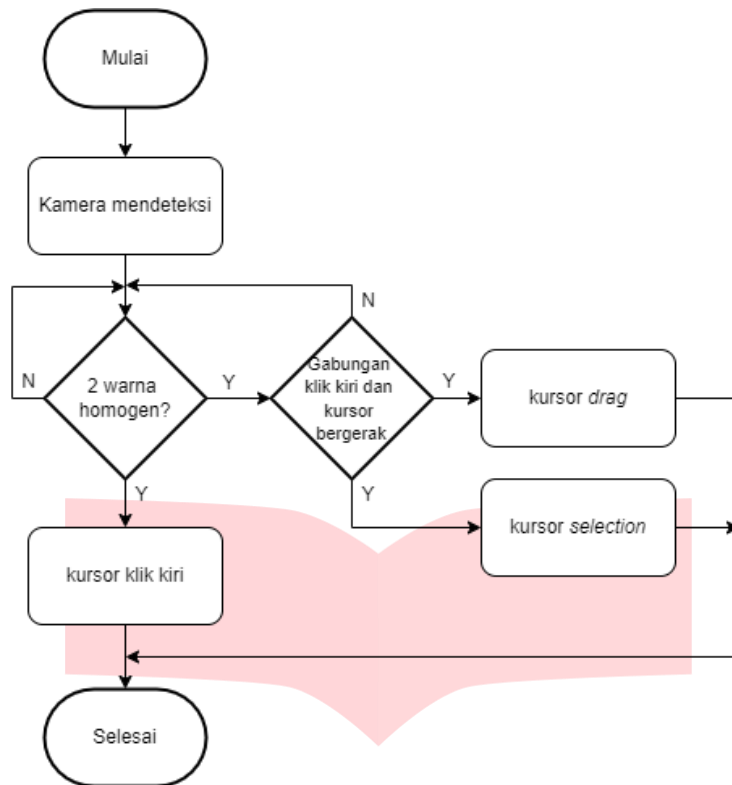
Dari Gambar 3.11 akan dijelaskan tahapan dan alur proses pada *flowchart* proses warna terdeteksi diatas, berikut penjelasannya.

- a. Dua warna homogen dipisah  
Setelah kamera dapat mendeteksi citra, pada program selanjutnya dilakukan pemilihan warna. Pada bagian ini dua warna citra yang sama dipisahkan maka aksi yang dilakukan ialah kursor bergerak.
- b. Dua warna homogen disatukan

- c. Dua warna heterogen  
Pada bagian ini bila dua warna citra yang sama disatukan maka kursor melakukan aksi klik kiri.
- Pada bagian ini bila salah satu dari dua warna citra berbeda dengan nilai warna citra yang telah dimasukkan, maka warna yang terdeteksi hanya warna yang sesuai dengan nilai warna yang dimasukkan, berikut contoh gambarannya.



GAMBAR 3. 12  
DUA WARNA HETEROGEN (MERAH MUDA DAN HIJAU)



GAMBAR 3. 13  
FLOWCHART FITUR TAMBAHAN KURSOR

Dari Gambar 3.13 akan dijelaskan tahapan dan alur proses pada *flowchart* fitur tambahan kursor, berikut penjelasannya.

1. **Kursor Drag**  
Fitur ini terjadi dikarenakan gabungan dari kursor klik kiri dengan kursor bergerak, dengan syarat harus ada objek yang hendak di pindahkan.
2. **Kursor selection**  
Fitur ini terjadi dikarenakan gabungan dari kursor klik kiri dengan kursor bergerak, dengan syarat tidak ada objek yang hendak di pindahkan. Sehingga *output* yang diberikan oleh kursor ialah *blue box*.

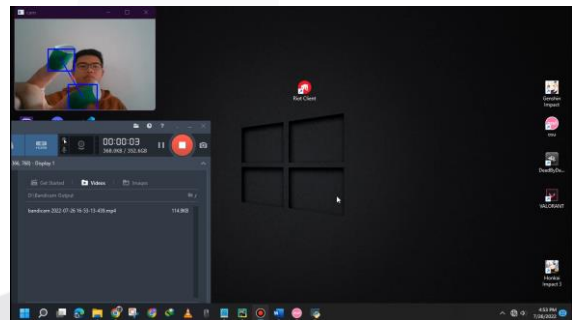
IV. HASIL DAN PEMBAHASAN

Hasil simulasi perancangan *virtual mouse* yang telah dilakukan berdasarkan beberapa parameter yang telah ditentukan, yaitu pengaruh warna, cahaya, dan jarak pengambilan objek menggunakan kamera *smartphone* dan *webcam*.

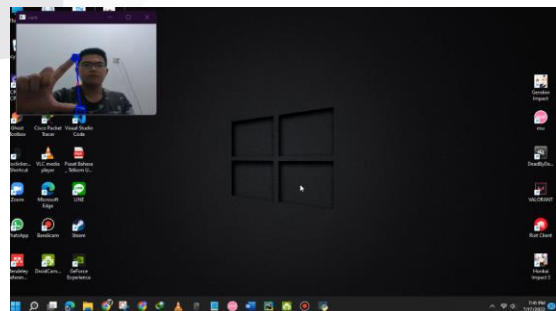
A. Skenario Pengujian

Pengujian dilakukan dengan cara menguji faktor jarak, kondisi cahaya (*indoor* dan *outdoor*), alat pada proses akuisisi (*smartphone* dan *webcam*). Saat melakukan pengujian antara *webcam* dengan warna citra dengan jarak 75 cm. Pengujian

berbentuk video menggunakan kamera *smartphone* dan laptop sebagai perbandingannya.



GAMBAR 4. 1  
PENGUJIAN MENGERAKAN KURSOR KONDISI OUTDOOR



GAMBAR 4. 2  
PENGUJIAN MENGERAKAN KURSOR KONDISI INDOOR

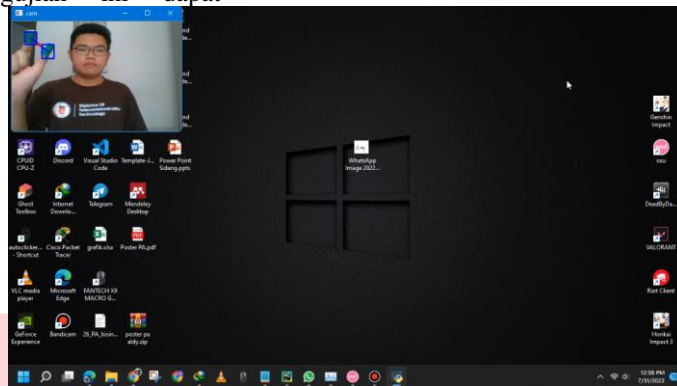
1. Pengujian *Outdoor*

Pada pengujian *virtual mouse* dilakukan di ruang tamu, pengujian

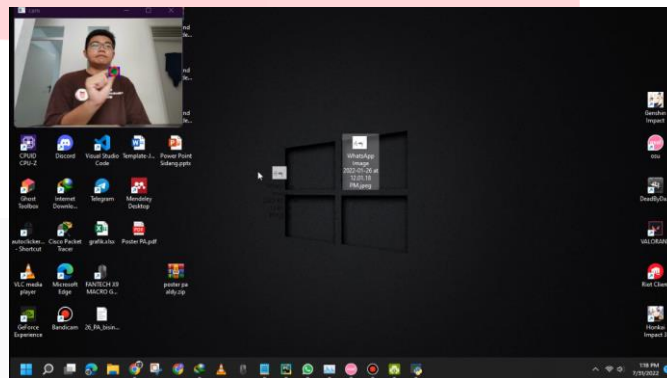


*outdoor* dilakukan saat siang hari pada pukul 12:00 - 14:00 WIB dan sore hari pada pukul 16:00 – 17:00 WIB karena index bias cahaya dari matahari menerangi ruangan tersebut. Dengan dilakukannya pengujian ini dapat

melihat tingkat keberhasilan dari kamera yang digunakan, sehingga dapat membandingkan kamera *smartphone* atau kamera laptop dengan akuisisi yang paling optimal.



GAMBAR 4.3  
PENGUJIAN KAMERA WEBCAM KONDISI *OUTDOOR*

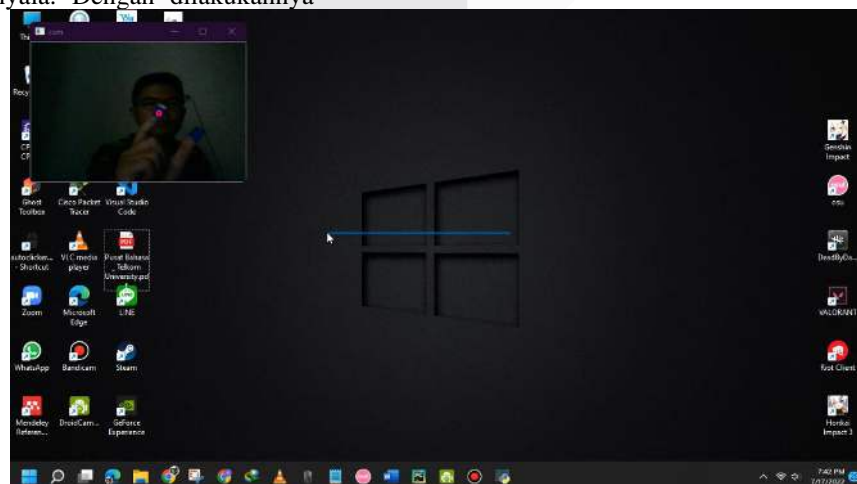


GAMBAR 4.4  
PENGUJIAN KAMERA *SMARTPHONE* KONDISI *OUTDOOR*

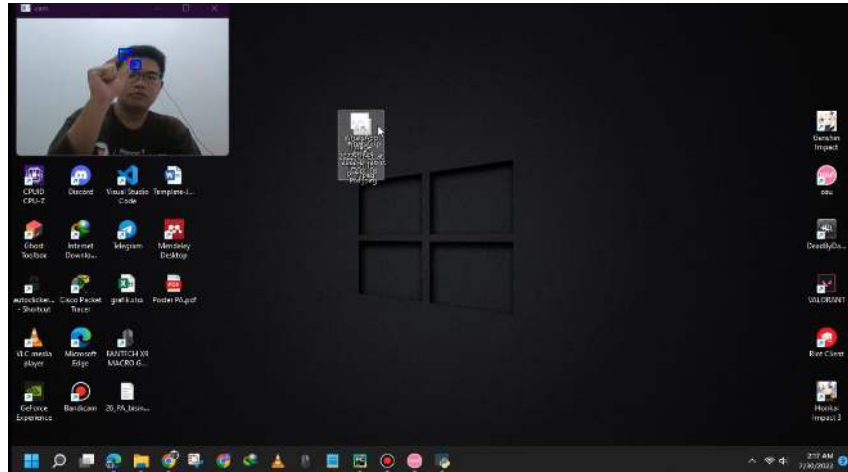
## 2. Pengujian *Indoor*

Selanjutnya pengujian *virtual mouse* dilakukan di kamar tidur, dengan sumber penerangan dari lampu. Pengujian *indoor* dibagi menjadi dua pengujian, yaitu pengujian dalam kondisi lampu mati dan dalam kondisi lampu menyala. Dengan dilakukannya

pengujian ini dapat melihat tingkat keberhasilan dari kamera yang digunakan, sehingga dapat membandingkan kamera *smartphone* atau kamera laptop dengan akuisisi yang paling optimal. Pengujian ini dilakukan malam hari pada pukul 19:00 – 22:00 WIB.



GAMBAR 4.5  
PENGUJIAN WEBCAM KONDISI *INDOOR* LAMPU MATI

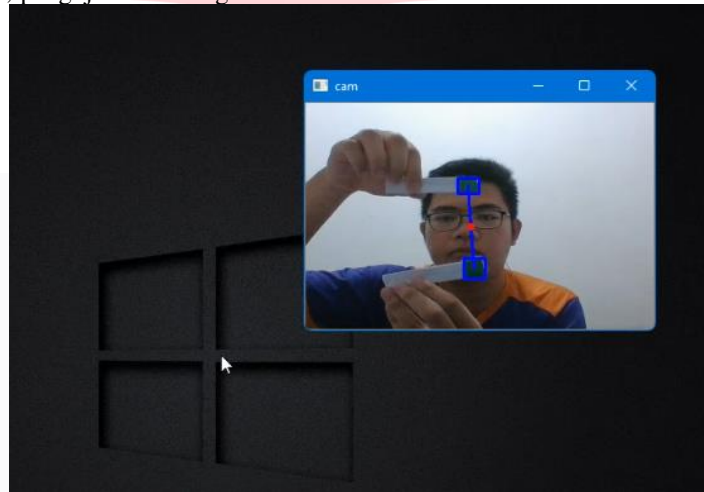


GAMBAR 4. 6  
PENGUJIAN WEBCAM KONDISI *INDOOR* LAMPU MENYALA

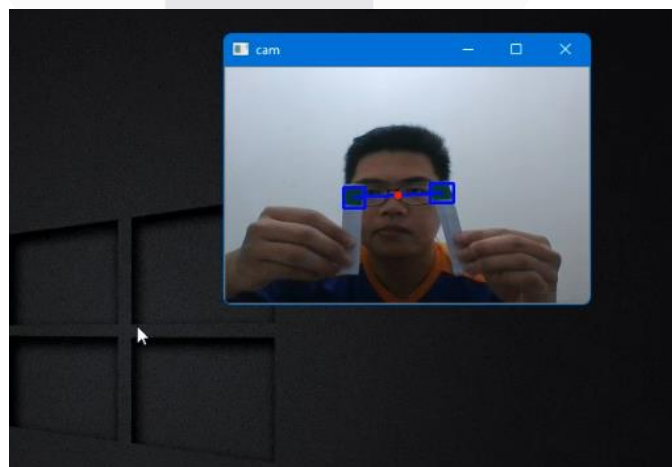
3. Pengujian posisi kursor

Selanjutnya dilakukan pengujian posisi pada *red dot*, pengujian ini dibagi

menjadi dua bagian, pertama pengujian vertikal dan kedua pengujian *horizontal*.



GAMBAR 4. 7  
PENGUJIAN VERTIKAL



GAMBAR 4. 8  
PENGUJIAN *HORIZONTAL*

Berdasarkan Gambar 4.7 dan Gambar 4.8 dapat disimpulkan dengan bermacam posisi apa bila *red*

*dot* masih diposisi yang sama, maka pergerakan kursor tidak berpengaruh.

### B. Pengujian Klik

Pada pengujian ini dilakukan deteksi sekaligus klik di beberapa warna yakni merah, hijau, kuning,

biru, dan merah muda. Berdasarkan hasil simulasi yang telah dilakukan terhadap lima warna tersebut didapatkan hasil pengujian citra sebagai berikut.

TABEL 4.1  
HASIL PENGUJIAN KLIK

NO.	Warna	Indoor		Outdoor		Akurasi
		Smartphone	Webcam	Smartphone	Webcam	
1.	Merah	✓	✓	✓	✓	100%
2.	Hijau	✓	✓	✓	✓	100%
3.	Biru	✓	✓	✓	✓	100%
4.	Merah muda	✓	✓	✓	✓	100%
5.	Kuning	✓	✓	✗	✗	50%
	Rata-rata	100%	100%	80%	80%	

Dari Tabel 4.1 menunjukkan bahwa dari lima warna yang sudah diujikan, warna kuning memiliki tingkat akuisisi paling rendah dalam kondisi *outdoor*. Warna merah, hijau, biru, dan merah muda memiliki tingkat akuisisi cukup baik dalam kondisi *indoor* maupun *outdoor*.

Pada pengujian ini dilakukan pendeteksian warna sekaligus jarak antara citra dengan kamera. Untuk pengujian ini batas jarak yang diukur 75 cm dari kamera, pembuktian dibagi menjadi 3 jarak pada titik 75 cm, 50 cm, dan 25 cm menggunakan warna merah, hijau, kuning, biru, dan merah muda. Berdasarkan hasil simulasi yang telah dilakukan terhadap lima warna tersebut didapatkan hasil pengujian citra sebagai berikut.

### C. Pengujian Jarak

TABEL 4.2  
HASIL DETEKSI WARNA DENGAN JARAK MAX 75 CM

NO.	Warna	Indoor		Outdoor		Akurasi
		Smartphone	Webcam	Smartphone	Webcam	
1.	Merah	✗	✓	✗	✗	25%
2.	Hijau	✗	✓	✗	✓	50%
3.	Biru	✗	✓	✗	✓	50%
4.	Merah muda	✗	✗	✗	✓	25%
5.	Kuning	✗	✗	✗	✗	0%
	Rata-rata	0%	60%	0%	60%	

TABEL 4.3  
HASIL DETEKSI WARNA DENGAN JARAK MAX 50 CM

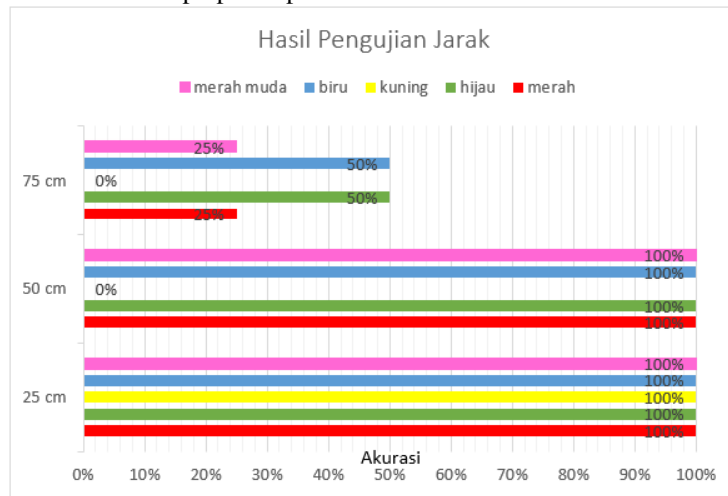
NO.	Warna	Indoor		Outdoor		Akurasi
		Smartphone	Webcam	Smartphone	Webcam	
1.	Merah	✓	✓	✓	✓	100%
2.	Hijau	✓	✓	✓	✓	100%
3.	Biru	✓	✓	✓	✓	100%
4.	Merah muda	✓	✓	✓	✓	100%
5.	Kuning	✗	✗	✗	✗	0%
	Rata-rata	80%	80%	80%	80%	

TABEL 4.3  
HASIL DETEKSI WARNA DENGAN JARAK MAX 25 CM

NO.	Warna	Indoor		Outdoor		Akurasi
		Smartphone	Webcam	Smartphone	Webcam	
1.	Merah	✓	✓	✓	✓	100%
2.	Hijau	✓	✓	✓	✓	100%
3.	Biru	✓	✓	✓	✓	100%
4.	Merah muda	✓	✓	✓	✓	100%
5.	Kuning	✓	✓	✓	✓	100%
	Rata-rata	100%	100%	100%	100%	

Berdasarkan pengujian jarak diperoleh tingkat akurasi tertinggi 100% untuk semua warna pada jarak 25 cm menggunakan kamera laptop maupun

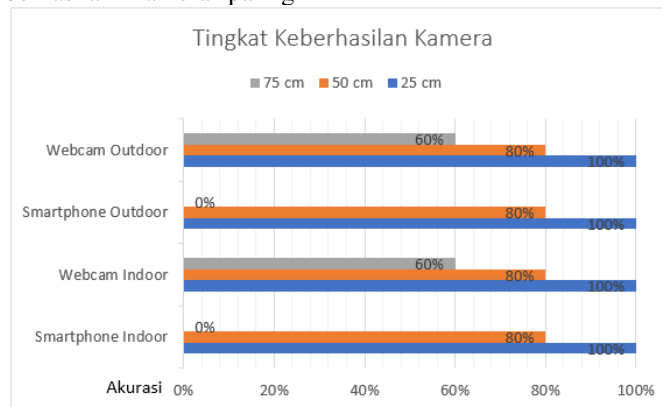
kamera *smartphone*. Sehingga dapat ditampilkan dalam bentuk grafik sebagai berikut.



GAMBAR 4.9  
GRAFIK HASIL PENGUJIAN JARAK 25 CM - 75 CM

Berdasarkan grafik pada Gambar 4.9 dapat disimpulkan tingkat keberhasilan kamera paling

tinggi adalah dengan penggunaan *webcam indoor* dan *outdoor*.

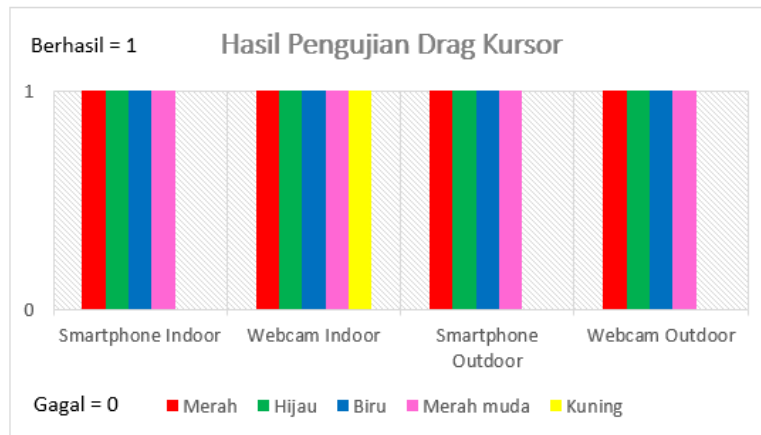


GAMBAR 4.10  
GRAFIK TINGKAT KEBERHASILAN DETEKSI KAMERA KONDISI *OUTDOOR* DAN *INDOOR*

Dari Tabel 4.4 menunjukkan tingkat keberhasilan deteksi kamera dan tingkat akurasi mencapai 100%, walaupun pada jarak 25 cm terdapat warna citra yang *glitch* setidaknya kamera dapat mendeteksi dalam kondisi *indoor* maupun *outdoor*. Dikarenakan banyaknya index cahaya yang masuk dalam kamera yang menyebabkan nilai dari warna citra yang sudah dimasukkan ke program berubah sehingga citra tidak dapat terdeteksi dengan baik.

D. Pengujian *Drag*

Pada pengujian ini dilakukan *drag* atau perpindahan objek menggunakan cursor dengan program *virtual mouse* yang telah dibuat. Untuk pengujian ini menggunakan kamera *smartphone* dan laptop, pembuktian menggunakan warna merah, hijau, kuning, biru, dan merah muda. Berdasarkan hasil simulasi yang telah dilakukan terhadap lima warna tersebut didapatkan hasil pengujian citra sebagai berikut.



GAMBAR 4. 11  
HASIL KEBERHASILAN DALAM MELAKUKAN DRAG KURSOR

Dari hasil Gambar 4.11 menunjukkan tingkat keberhasilan bernilai 1 dan gagal bernilai 0 dalam melakukan *drag* atau memindahkan *file/aplikasi* dengan jarak 5 cm – 75 cm. Saat melakukan pengujian sering terdapat *glitch* pada jarak lebih dari 60 cm. Itu dikarenakan tingkat akurasi dari index bias cahaya dan jarak kamera dengan citra. Hasil yang telah ditunjukkan pada gambar di atas warna kuning memiliki tingkat akurasi dan keberhasilan yang kurang baik dibandingkan dengan 4 warna lainnya. Sehingga bisa disimpulkan untuk melakukan *drag* atau memindahkan *file/aplikasi* jarak maksimum 75 cm.

E. Pengujian Jarak *Selection*

Pada pengujian ini dilakukan pemilihan atau *selection* pada objek menggunakan kursor dengan program *virtual mouse* yang telah dibuat. Untuk pengujian ini menggunakan kamera *smartphone* dan laptop, pembuktian menggunakan warna merah, hijau, kuning, biru, dan merah muda. Berdasarkan hasil simulasi yang telah dilakukan terhadap lima warna tersebut didapatkan hasil pengujian citra sebagai berikut.

TABEL 4. 4  
HASIL KEBERHASILAN DALAM MELAKUKAN SELECTION DENGAN JARAK 75 CM

NO.	Warna	Indoor		Outdoor		Akurasi
		Smartphone	Webcam	Smartphone	Webcam	
1.	Merah	✗	✗	✗	✗	0%
2.	Hijau	✗	✓	✗	✗	25%
3.	Biru	✗	✓	✗	✓	50%
4.	Merah muda	✗	✗	✗	✗	0%
5.	Kuning	✗	✗	✗	✗	0%
	Rata-rata	0%	50%	0%	20%	

TABEL 4. 5  
HASIL KEBERHASILAN DALAM MELAKUKAN SELECTION DENGAN JARAK 50 CM

NO.	Warna	Indoor		Outdoor		Akurasi
		Smartphone	Webcam	Smartphone	Webcam	
1.	Merah	✓	✓	✓	✓	100%
2.	Hijau	✓	✓	✓	✓	100%
3.	Biru	✓	✓	✓	✓	100%
4.	Merah muda	✓	✓	✓	✓	100%
5.	Kuning	✗	✗	✗	✗	0%
	Rata-rata	80%	80%	80%	80%	

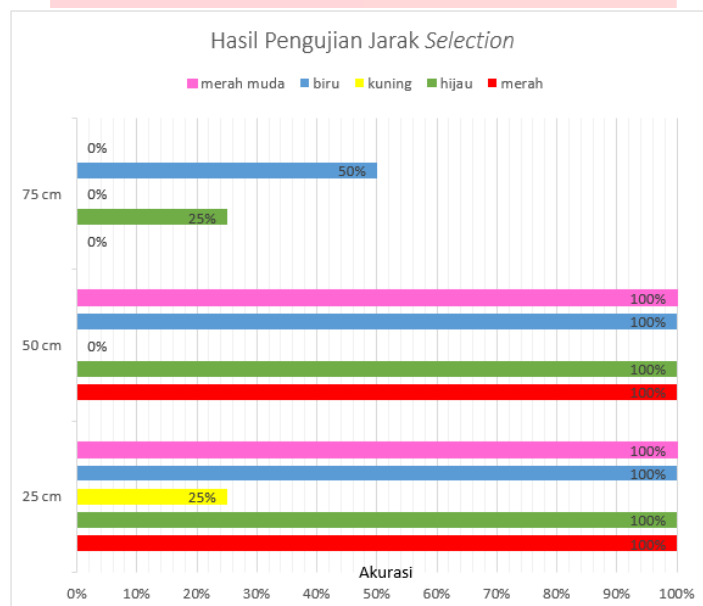
Tabel 4. 6 Hasil keberhasilan dalam melakukan *selection* dengan jarak 25 cm

NO.	Warna	Indoor		Outdoor		Akurasi
		Smartphone	Webcam	Smartphone	Webcam	
1.	Merah	✓	✓	✓	✓	100%
2.	Hijau	✓	✓	✓	✓	100%
3.	Biru	✓	✓	✓	✓	100%
4.	Merah muda	✓	✓	✓	✓	100%
5.	Kuning	✗	✓	✗	✗	25%
	Rata-rata	80%	100%	80%	80%	

Fitur *Selection* pada kursor terjadi karena adanya 2 aksi yang mana klik dan *drag*, disebabkan tidak ada objek yang hendak dipindahkan atau digeser. Pada tabel 4.5 menunjukkan pada jarak 75 cm menggunakan kamera *smartphone* warna citra tidak terbaca baik *indoor* maupun *outdoor*. Sedangkan kamera laptop untuk *indoor* memiliki

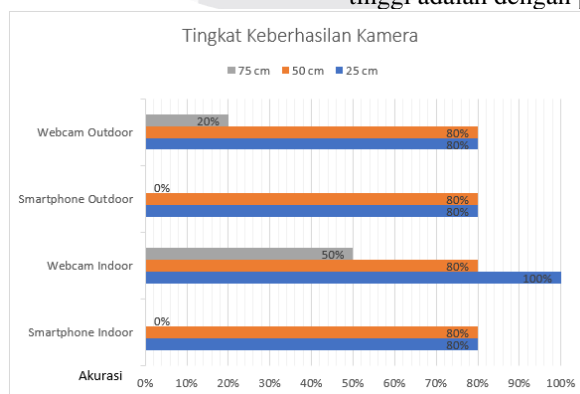
persentase 50% dan *outdoor* memiliki persentase 20%.

Tabel 4.6 menunjukkan pada jarak 50 cm menggunakan kamera *smartphone* dan laptop warna citra yang dapat terdeteksi meningkat menjadi 80%. Lalu tabel 4.7 menunjukkan pada jarak 25 cm dalam kondisi *indoor* tingkat keberhasilan pada kamera laptop 100% selebihnya 80%. Berikut grafik pengujian *selection*.



GAMBAR 4. 12  
GRAFIK HASIL PENGUJIAN JARAK SELECTION 25 CM - 75 CM

Berdasarkan grafik pada Gambar 4.13 dapat disimpulkan tingkat keberhasilan kamera paling tinggi adalah dengan penggunaan *webcam indoor*.



GAMBAR 4. 13  
GRAFIK TINGKAT KEBERHASILAN DETEKSI SELECTION KAMERA KONDISI OUTDOOR DAN INDOOR

## V. KESIMPULAN

Pada proyek akhir ini telah dirancang sebuah sistem *virtual mouse* berbasis *image processing* dengan *library opencv* menggunakan bahasa pemrograman *python*. Pengujian yang telah dilakukan, yaitu fitur klik, *drag*, dan *selection*. Adapun hasil yang diperoleh adalah sebagai berikut.

1. Pengujian jarak kursor dibagi menjadi 3 bagian, yaitu 25 cm, 50 cm, dan 75 cm. Didapatkan rata-rata tingkat akurasi terbaik 100% pada jarak 25 cm, berwarna merah, kuning, hijau, biru, dan merah muda. Sehingga jarak yang paling optimal pada pengujian jarak kursor ialah 25 cm.
2. Pengujian klik kursor didapatkan rata-rata tingkat akurasi terbaik 90% dengan tingkat akurasi tertinggi 100% pada warna merah, hijau, biru, dan merah muda.
3. Pengujian jarak *selection* kursor dibagi menjadi 3 bagian, yaitu 25 cm, 50 cm, dan 75 cm. Didapatkan rata-rata tingkat akurasi terbaik 85% pada jarak 25 cm, dengan tingkat akurasi tertinggi 100% pada warna merah, hijau, biru, dan merah muda. Sehingga jarak yang paling optimal pada pengujian jarak *selection* kursor ialah 25 cm.
4. Pengujian *drag* kursor didapatkan tingkat keberhasilan terbaik pada kamera laptop/webcam kondisi *indoor* 100%. Sehingga jarak yang paling optimal pada pengujian jarak *drag* kursor ialah 10 - 60 cm.
5. Berdasarkan dari hasil percobaan dan perbandingan yang telah diuji. Dapat disimpulkan *virtual mouse* sangat dipengaruhi oleh nilai ambang batas dari setiap warna, tingkat kecerahan cahaya, dan jarak. Dengan begitu kursor dapat bergerak dengan lancar tanpa adanya *glitch*.

Thakare, "IJARCCE Virtual Mouse," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 5, no. 5, 2016, doi: 10.17148/IJARCCE.2016.5511.

- [4] R. Sande, N. Marathe, N. Bhegade, A. Lugade, and S. S. Jogdand, "Virtual Mouse using Hand Gestures," *Int. J. Adv. Res. Sci. Eng. Technol.*, vol. 8, 2021, [Online]. Available: [www.ijarset.com](http://www.ijarset.com).
- [5] Y. Erliani and B. Priambodo, "DETEKSI MATA REAL TIME MENGGUNAKAN OPENCV UNTUK ANDROID," 2015. [Online]. Available: <http://opencv.org>.
- [6] A. Riani Putri Jurusan Pendidikan Teknologi Informasi and S. PGRI Tulungagung Jl Mayor Sujadi Timur no, "PENGOLAHAN CITRA DENGAN MENGGUNAKAN WEB CAM PADA KENDARAAN BERGERAK DI JALAN RAYA," 2016.
- [7] M. Hermansyah and N. Fadillah, "Virtual Mouse Berdasarkan Warna RGB Menggunakan Metode Optical Flow Secara Real-Time," vol. 21, no. 2, pp. 249–252, 2019, doi: 10.31294/p.v20i2.
- [8] R. D. Kusumanto, A. N. Tompunu, D. Wahyu, and S. Pambudi, "Klasifikasi Warna Menggunakan Pengolahan Model Warna HSV," 2011.
- [9] "PENGENALAN BAHASA PEMROGRAMAN PYTHON MENGGUNAKAN APLIKASI GAMES UNTUK SISWA/I DI WILAYAH KEMBANGAN UTARA." [Online]. Available: <https://codecombat.com/>.

## REFERENSI

- [1] C. Halim and H. Prasetyo, "PENERAPAN ARTIFICIAL INTELLIGENCE DALAM COMPUTER AIDED INSTRUCTURE(CAI)," 2018.
- [2] K. Prabahar, "Virtual Mouse Using Hand Gesture Recognition," *SJ Impact Factor6*, vol. 887, 2017, [Online]. Available: [www.ijraset.com](http://www.ijraset.com)405.
- [3] P. Vaidya, P. Pangavhane, J. Deore, and R.