



UNIVERSITAS  
**Dinamika**

**SISTEM *AUTOMATIC FEATURE SELECTION* BERBASIS DETEKSI  
GESTUR KEDUA JARI TANGAN UNTUK MENGONTROL *LEVEL*  
KECEPATAN PUTARAN 2 KIPAS ANGIN MENGGUNAKAN  
MEDIAPIPE**



**LAPORAN TUGAS AKHIR**

**Program Studi**

**S1 Teknik Komputer**

UNIVERSITAS  
**Dinamika**

**Oleh:**

**Yumerius Rafael Biidapode Edowai**

**19410200039**

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2023**

**SISTEM *AUTOMATIC FEATURE SELECTION* BERBASIS DETEKSI  
GESTUR KEDUA JARI TANGAN UNTUK MENGONTROL *LEVEL*  
KECEPATAN PUTARAN 2 KIPAS ANGIN MENGGUNAKAN  
MEDIAPIPE**

**TUGAS AKHIR**

**Diajukan Sebagai salah satu syarat untuk mengerjakan  
Program Sarjana Teknik**



UNIVERSITAS  
**Dinamika**

**Disusun Oleh:**

**Nama : Yumerius R. B. Edowai**

**NIM : 19410200039**

**Program : S1 (Strata Satu)**

**Jurusan : Teknik Komputer**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2023**

## TUGAS AKHIR

### SISTEM *AUTOMATIC FEATURE SELECTION* BERBASIS DETEKSI GESTUR KEDUA JARI TANGAN UNTUK MENGONTROL *LEVEL* KECEPATAN PUTARAN 2 KIPAS ANGIN MENGGUNAKAN MEDIAPIPE

Dipersiapkan dan disusun oleh:  
**Yumerius Rafael Biidapode Edowai**  
NIM : 19410200039

Telah diperiksa, diuji, dan disetujui oleh  
Dewan Pembahas Pada: 17 Juli 2023

#### Susunan Dewan Pembahas

##### Pembimbing:

- I. Heri Pratikno, M.T., MTCNA., MTCRE.  
NIDN 0716117302
- II. Weny Indah Kusumawati, S.Kom., M.MT  
NIDN 0721047201



Digitally signed by  
Heri Pratikno, M.T.  
Date: 2023.08.01  
11:40:38 +07'00'



Universitas  
Dinamika  
2023.08.01  
09:20:26 +07'00'

##### Pembahas:

Musayyanah, S.ST., M.T  
NIDN 0730069102



Digitally signed by Musayyanah  
DN: cn=Musayyanah,  
o=Universitas Dinamika, ou=SI  
Teknik Komputer,  
email=musayyanah@dinamika.ac  
-id, c=ID  
Date: 2023.08.01 17:38:22 +07'00'  
Adobe Acrobat Reader version:  
2023.003.20244

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana



Digitally signed by  
Universitas Dinamika  
Date: 2023.08.02  
11:04:58 +07'00'

**Tri Sagirani, S.Kom., M.MT.**

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

*“Jangan menyerah, terus melangkah”*



UNIVERSITAS  
**Dinamika**

Terima kasih banyak kepada kedua orangtua saya dan keluarga besar HMTK S1  
Teknik Komputer yang selalu senantiasa mendoakan dan selalu mensupport saya.



UNIVERSITAS  
**Dinamika**

## PERNYATAAN

### PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya:

Nama : Yumerius Rafael Biidapode Edowai  
NIM : 19410200039  
Program Studi : S1 Teknik Komputer  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Tugas Akhir  
Judul Karya : **SISTEM AUTOMATIC FEATURE SELECTION BERBASIS DETEKSI GESTUR KEDUA JARI TANGAN UNTUK MENGONTROL LEVEL KECEPATAN PUTARAN 2 KIPAS ANGIN MENGGUNAKAN MEDIAPIPE**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberika kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (*Database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, buka plagiat baik sebagai maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pecabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikian surat ppernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 15 Juni 2023



Yumerius Rafael Biidapode Edowai

NIM: 19410200039

## ABSTRAK

Perkembangan teknologi telah mendorong manusia untuk mempermudah pekerjaan. Perkembangan teknis juga mengalami kemajuan yang sangat cepat. Pesatnya perkembangan teknologi di era globalisasi saat ini telah membawa banyak keuntungan bagi kemajuan dari berbagai sudut pandang. Mikrokontroler saat ini menikmati perkembangan pesat dan direkomendasikan dalam berbagai aplikasi kontrol. Salah satu mikrokontroler yang paling umum adalah Arduino. Arduino adalah mikrokontroler, yaitu sistem komputer yang berjalan pada sebuah chip. *Computer Vision* adalah salah satu cabang ilmu dan kecerdasan komputer yang bertujuan untuk melihat dan memahami benda atau objek yang ada disekitarnya. Penelitian Tugas Akhir ini mengkaji implementasi *framework* Mediapipe sebagai cara untuk mengontrol kecepatan putar kipas menggunakan Arduino melalui proses pengenalan berdasarkan bentuk jari kedua tangan, sedangkan *finger gesture* yang akan diimplementasikan hanya melibatkan tiga jari dengan jari kepal, dari masing-masing kedua tangan. Pada tabel jarak hasil rata-rata nilai akurasi semua deteksi gestur jari yang dilakukan dengan metode Mediapipe pada jarak 50cm sampai 250cm, dilakukan 3 kali percobaan pada setiap jarak, pada tangan kanan tingkat akurasi gestur jari 0 sebesar 86,67%, akurasi gestur 1 sebesar 86,67%, akurasi gestur 2 sebesar 100%, akurasi gestur 3 sebesar 93,33 % dan pada tangan kiri tingkat akurasi gestur jari 0 sebesar 86,67%, akurasi gestur 1 sebesar 100%, akurasi gestur 2 sebesar 93,33%, dan akurasi gestur 3 sebesar 93,33%. Pada percobaan ini juga menghasilkan nilai rata-rata FPS untuk setiap gestur jari tangan dengan rata-rata gestur jari 0 sebesar 34.36, gestur jari 1 sebesar 34.05, gestur jari 2 sebesar 33.15 dan gestur jari 3 sebesar 33.55.

Kata Kunci: Mediapipe, Arduino, *Computer Vision*, *Automatic Feature Selection*

## KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa atas segala nikmat-Nya sehingga penulis dapat menyelesaikan laporan akhir proyek yang berjudul “Sistem Automatic Feature Selection Berbasis Deteksi Gestur Kedua Jari Tangan Untuk Mengontrol Level Kecepatan Putaran 2 Kipas Angin Menggunakan Mediapipe”. Selama mengerjakan laporan akhir ini, penulis banyak mendapat bantuan dari berbagai sumber. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Tuhan Yesus, karena dengan Rahmat-Nya penulis dapat menyelesaikan Laporan Tugas Akhir ini.
2. Orang tua dan seluruh keluarga yang telah memberikan dorongan dan dukungan baik secara moril maupun materil, sehingga penulis dapat menempuh dan menyelesaikan Laporan Tugas Akhir ini.
3. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
4. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
5. Ibu Musayyanah, S.ST., M.T., selaku dosen pembahas. Penulis mengucapkan terimakasih atas bimbingan dan kesempatan serta tuntunan baik secara tertulis, sehingga penulis dapat menyelesaikan Tugas Akhir.
6. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku dosen pembimbing pertama saya telah memberikan dukungan berupa motivasi serta saran agar Laporan Tugas Akhir ini berjalan dengan lancar.
7. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku dosen pembimbing kedua yang banyak membantu agar Laporan Tugas Akhir ini berjalan dengan lancar.
8. Semua teman-teman S1 Teknik Komputer Angkatan 2019 yang sudah memberikan masukan dan doa serta semangat dalam Laporan Tugas Akhir ini.
9. Semua pihak yang tidak dapat disebutkan satu per satu yang telah memberikan dukungan dan bantuan dalam bentuk apapun dalam penyusunan laporan akhir ini.



Penulis berharap semoga laporan ini dapat membantu dan bermanfaat dalam memberikan informasi kepada para pembaca. Penulis juga menyadari masih banyak kekurangan dalam penulisan laporan ini. Oleh karena itu penulis sangat mengharapkan saran dan kritik untuk memperbaiki kekurangan dan menjadi lebih baik lagi.

Surabaya, 17 Juli 2023

Penulis

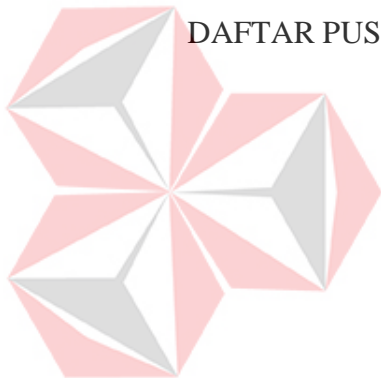


UNIVERSITAS  
**Dinamika**

## DAFTAR ISI

	Halaman
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI .....	x
DAFTAR GAMBAR .....	xii
DAFTAR TABEL .....	xiii
DAFTAR LAMPIRAN .....	xiv
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Manfaat .....	4
BAB II LANDASAN TEORI .....	5
2.1 OpenCV .....	5
2.2 Python .....	5
2.3 Mediapipe .....	6
2.4 Arduino IDE .....	6
2.5 Arduino Uno .....	7
2.6 Modul Relay .....	7
BAB III METODOLOGI PENELITIAN .....	9
3.1 Instalasi <i>Environment</i> .....	9
3.2 Perancangan Perangkat Keras .....	9
3.3 Dataset .....	10
3.4 Metode Transmisi Serial .....	12
3.5 Metode Deteksi Dengan Mediapipe .....	12
3.6 Proses Testing Program Python .....	13
3.7 <i>Flowchart</i> Program Mikrokontroler .....	14
3.8 Komunikasi Serial antara <i>Computer Vision</i> dan Mikrokontroler Arduino .....	14

3.8.1 Tujuan Komunikasi Serial antara <i>Computer Vision</i> dan Mikrokontroler Arduino .....	14
3.8.2 Prosedur Membangun Komunikasi serial antara <i>Computer Vision</i> dan Mikrokontroler Arduino.....	14
3.8.3 Hasil dari Membangun Komunikasi serial antara <i>Computer Vision</i> dan Mikrokontroler Arduino .....	14
BAB IV HASIL DAN PEMBAHASAN .....	19
4.1 Pengujian Metode Mediapipe.....	19
4.1.1 Prosedur Pengujian Metode Mediapipe .....	19
4.1.2 Keypoint Gestur Jari Kedua Tangan .....	20
4.1.3 Hasil Pengujian Metode Mediapipe .....	20
BAB V PENUTUP .....	26
5.1 Kesimpulan.....	26
5.2 Saran .....	26
DAFTAR PUSTAKA .....	28



UNIVERSITAS  
**Dinamika**

## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Contoh penerapan OpenCV .....	5
Gambar 2.2 Bahasa pemograman Python .....	6
Gambar 2.3 Bentuk hand gestur Mediapipe .....	6
Gambar 2.4 Arduino Uno .....	7
Gambar 2.5 Modul Relay 4 Channel.....	8
Gambar 3.1 Alur perancangan <i>Hardware</i> .....	9
Gambar 3.2 Skema <i>Hardware</i> .....	10
Gambar 3.3 Dataset gestur jari tangan .....	11
Gambar 3.4 <i>Flowchart</i> proses training dataset .....	12
Gambar 3.5 <i>Flowchart</i> proses testing Program Python .....	15
Gambar 3.6 <i>Flowchart</i> proses testing program .....	15
Gambar 3.7 Gagal dalam komunikasi serial .....	17
Gambar 3.8 Python dapat mengeluarkan nilai .....	17
Gambar 3.9 Arduino dapat mengunggah program.....	18
Gambar 4.1 Keypoint jari kedua tangan.....	20
Gambar 4.2 Hasil deteksi jari kedua tangan.....	20
Gambar 4.3 Hasil deteksi gestur jari untuk mengontrol kecepatan kipas.....	25

## DAFTAR TABEL

	Halaman
Tabel 4.1 Tabel skema <i>Hardware</i> .....	10
Tabel 4.2 Hasil uji jarak jari kedua tangan .....	21
Tabel 4.3 Hasil pengujian deteksi gestur jari kedua tangan .....	23
Tabel 4.4 Hasil pengujian kontrol kecepatan kipas angin.....	25



UNIVERSITAS  
**Dinamika**

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Listing Program Python.....	31
Lampiran 2 Listing Program Arduino Uno .....	34



UNIVERSITAS  
**Dinamika**

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Perkembangan teknologi telah mendorong manusia untuk mempermudah pekerjaan. Perkembangan ini juga mengalami kemajuan yang sangat cepat. Pesatnya perkembangan teknologi di era globalisasi saat ini telah membawa banyak keuntungan dari berbagai segi. Perkembangan ini selalu berkat upaya orang-orang yang ingin tahu dan mau, semuanya menjadi lebih praktis. Saat ini orang tidak ingin melakukan apapun dan melakukan semuanya dengan cepat tanpamenghabiskan banyak energi dan membuat pekerjaan lebih mudah, (Arisandi, 2014). Oleh karena itu, mikrokontroler dipandang sebagai salah satu bentuk teknologi yang memudahkan pekerjaan manusia.

*Automatic Feature Selection* menjadi salah satu landasan dalam mendeteksi bentuk gestur jari tangan. Sehingga perlu mengumpulkan data yang mencakup gestur jari kedua tangan. Data ini bisa berupa citra atau video yang menampilkan tangan dan jari-jarinya dalam berbagai posisi dan gerakan. Oleh karena itu perlu mengekstraksi fitur-fitur yang relevan dari data tersebut. Fitur-fitur ini bisa meliputi posisi jari, sudut antara jari-jari, panjang jari, atau karakteristik lainnya yang dapat membedakan gestur jari yang berbeda.

Mikrokontroler saat ini berkembang pesat dan semakin dicari untuk aplikasi sistem kontrol. Salah satu modul mikrokontroler yang paling umum adalah Arduino. Arduino adalah mikrokontroler, yang merupakan sistem komputer dalam sebuah *chip*. Arduino juga memiliki memori, prosesor, I/O dan mikrokontroler, dapat dikatakan merupakan versi komputer mini dengan perangkat lunak pendukung pemrograman Arduino IDE.

*Computer Vision* merupakan cabang ilmu dan kecerdasan komputer yang bertujuan untuk melihat dan memahami benda atau objek yang ada disekitarnya. Kemampuan ini memungkinkan komputer mampu menganalisis target itu sendiri untuk informasi perintah. Salah satu jenis CV (*Computer Vision*) adalah *OpenCV*. *OpenCV* (Open Computer Vision) adalah perangkat lunak library open-source punya kegunaanya sendiri contohnya deteksi benda, deteksi wajah, pengenalan

objek, deteksi gestur jari tangan, dan lainnya.

Saat ini sudah banyak Arduino yang digunakan dalam sistem kontrol sehingga dapat memenuhi keperluan manusia, namun masih belum banyak diimplementasikan untuk Computer Vision dalam berbagai hal. Meski sekarang ini *computer vision* sudah bisa 3 dipadukan dengan mikrokontroler. Hal ini digunakan sebagai peluang untuk menemukan solusi untuk masalah tersebut. Selain itu, pengimplementasian teknologi menggunakan framework Mediapipe dapat berjalan secara efektif sebagai alat untuk mendeteksi gerakan tangan yang kompleks secara akurat. Karena kemampuannya fitur ekstraksi landmark dari Mediapipe cocok dengan kemampuannya untuk mendeteksi gesture tangan, seperti pada penelitian tahun 2022 oleh Muhammad Rifki Pratama Nautica yang berjudul Hand Gestur Detection sebagai alat bantu ajar behitung menggunakan Mediapipe dan Convolution Neural Network secara realtime (Nautica, 2022), serta juga untuk melanjutkan apa yang sudah di peneliti untuk oleh M. Aldi Fakhruddin tentang Sistem Deteksi Gestur Jari Tangan Menggunakan Mediapipe Dan FasterRCNN Untuk Mengontrol Kecepatan Kipas Angin, (Fakhruddin, 2022).

Pada penelitian Tugas Akhir sebelumnya, yang telah dilakukan oleh (Nautica, 2022) adalah menghitung kesepuluh jari tangan secara otomatis menggunakan mediapipe yang hasil perhitungannya ditampilkan dilayar. Pada penelitian (Fakhruddin, 2022) mengimplementasikan framework Mediapipe dan Faster-RCNN sebagai alat pengontrol kecepatan putaran hanya 1 kipas angin menggunakan Arduino melalui proses deteksi gestur jari tangan kanan atau deteksi gestur jari tangan kiri saja.

Pada Tugas Akhir ini, penulis membuat sistem yang dapat mengontrol level kecepatan putaran dua kipas angin berbasis deteksi bentuk gestur jari tangan kanan dan kiri. Dalam memilih pengontrolan kedua kipas angin tersebut dapat dilakukan secara otomatis berdasarkan seleksi fitur gestur tangan kanan dan tangan kiri menggunakan mediapipe. Level kecepatan putaran kipas angin 1 hanya dapat dikontrol oleh tangan kanan saja, sedangkan level kecepatan putaran kipas angin 2 dapat dikontrol oleh tangan kiri. Pengontrolan kedua kipas angin tersebut dapat dilakukan secara bersamaan yang mana antara kipas angin 1 dengan kipas angin 2 tidak saling berkaitan.



## 1.2 Rumusan Masalah

Dari latar belakang diatas, maka dapat dirumuskan masalah pada Tugas Akhir ini sebagai berikut:

1. Bagaimana mengontrol *level* kecepatan 2 kipas angin secara *computer vision* melalui deteksi bentuk gesture jari kedua tangan?
2. Bagaimana tingkat *realtime* pada sistem seleksi fitur bentuk gestur jari kedua tangan secara otomatis menggunakan Mediapipe?
3. Berapa besar tingkat akurasi sistem seleksi fitur untuk deteksi bentuk gestur jari kedua tangan?
4. Berapa jarak yang memiliki akurasi tertinggi antara webcam dan jari kedua tangan?

## 1.3 Batasan Masalah

Pada pembuatan penelitian ini, batasan masalah dibatasi pada beberapa hal berikut:

1. Kipas angin 1 hanya dapat dikontrol oleh jari tangan kanan.
2. Kipas angin 2 hanya dapat dikontrol oleh jari tangan kiri.
3. Pencahayaan yang merata pada ruangan.
4. *Level* kecepatan putaran masing-masing kipas angin hanya dapat dikontrol melalui empat bentuk gestur jari tangan.

## 1.4 Tujuan

Berdasarkan latar belakang dan rumusan masalah diatas, yang menjadi tujuan pada Tugas Akhir ini sebagai berikut:

1. Dapat mengontrol *level* kecepatan 2 kipas angin secara *computer vision* melalui deteksi bentuk gesture jari kedua tangan.
2. Dapat mengetahui tingkat *realtime* pada sistem seleksi fitur bentuk gestur jari kedua tangan secara otomatis menggunakan Mediapipe.
3. Mampu mengetahui tingkat akurasi sistem seleksi fitur untuk deteksi bentuk gestur jari kedua tangan.

4. Dapat mengetahui jarak yang memiliki tingkat akurasi tertinggi antara webcam dan jari kedua tangan.

### 1.5 Manfaat

Adapun madaat dari penelitian ini dapat diperoleh sebagai berikut:

1. Memudahkan manusia dalam mengontrol kecepatan putara kipas angin hanya dengan gesture jari tangan.
2. Interaksi pengontrolan kedua kipas angin dapat dilakukan lebih natural karena tidak perlu ada kontak fisik.
3. Memudahkan mahasiswa untuk mengembangkan pengetahuan tentang *computer vision* dengan Arduino sebagai antarmuka.
4. Adanya kecedasan buatan pada *computer vision* yang dapat mendeteksi perbedaan bentuk gestur jari kedua tangan dalam mengontrol *level* kecepatan ke - 2 kipas angin.



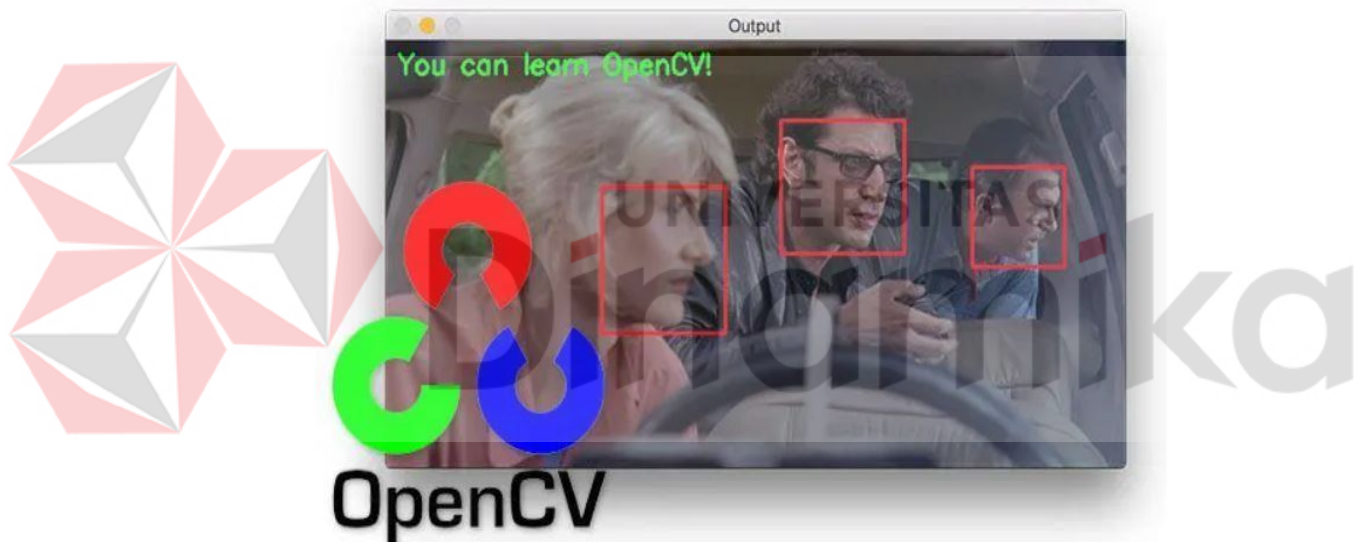
UNIVERSITAS  
**Dinamika**

## BAB II

### LANDASAN TEORI

#### 2.1 OpenCV

*OpenCV (Open Computer Vision Library)* adalah library perangkat lunak bersifat open-source yang memiliki lisensi BSD-Licensed product. *OpenCV* memiliki lebih dari 2500 jenis algoritma yang sudah teroptimasi dan tersedia untuk memenuhi kebutuhan mengenai *Computer Vision* dan *Machine Learning*. Berbagai macam algoritma *OpenCV* memiliki kegunaan masing-masing, seperti mendeteksi dan mengenali wajah, mendeteksi gerakan tangan, indentifikasi objek dan lain-lain (Ramdhon, 2021).



Gambar 2.1 Contoh penerapan OpenCV  
(Sumber: Andrian Rosebrock, 2018)

#### 2.2 Python

Python adalah sebuah bahasa pemrograman interpretatif yang mampu diterapkan dalam berbagai kebutuhan programming dengan filosofi rancangan yang 6 terfokus pada tingkat keterbacaan kode dan juga salah satu bahasa yang sering digunakan dalam bidang *Machine Learning*, *Data Science* dan *Internet of Things* (Zein, 2018). Pada penelitian ini, bahas pemrograman python yang akan digunakan sebagai bahasa utama untuk melakukan proses training dataset dan proses fitur selection untuk deteksi gesture kedua jari tangan.



Gambar 2.2 Bahasa pemrograman Python  
(Sumber: Logos-World)

### 2.3 Mediapipe

Mediapipe adalah kerangka kerja yang dikembangkan oleh Google yang memungkinkan untuk membangun pipeline untuk memproses data pengamatan dari berbagai macam format audio dan video. Mediapipe menargetkan bagi yang ingin menerapkan kecerdasan buatan pada aplikasi yang akan dibangun. Google telah menggunakan Mediapipe untuk penggunaan internal sejak 2012 dan menjadikannya open source pada tahun 2019. *Framework* ini menyediakan berbagai solusi *machine learning* seperti pengenalan wajah, holistic dan lainnya yang dapat dilihat pada web resmi Mediapipe. Solusi yang ditawarkan kompatibel dengan sistem open Android dan iOS, serta dengan bahasa C++, Python, JS dan Coral (Mediapipe, 2019).



Gambar 2.3 Bentuk hand gestur Mediapipe  
(Sumber: Mediapipe Hand 2006)

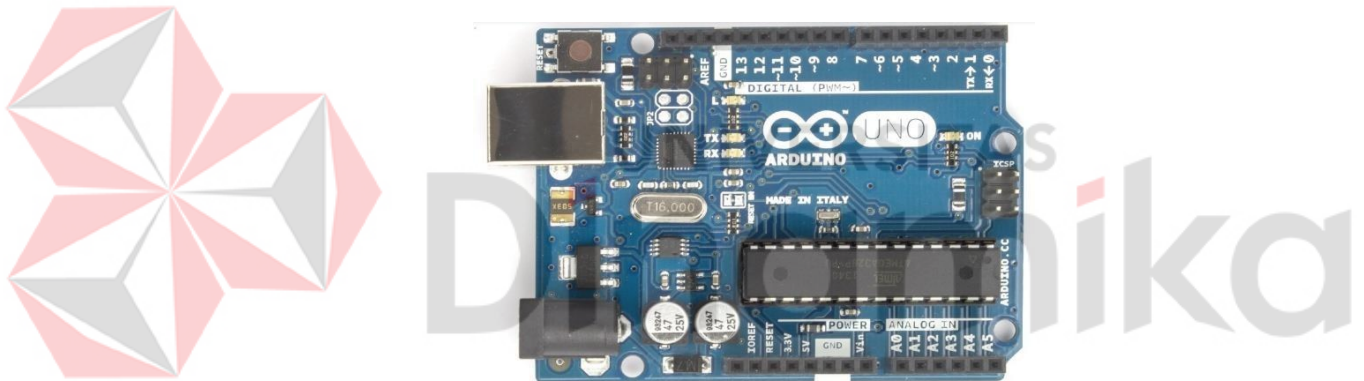
### 2.4 Arduino IDE

Arduino IDE (*Integrated Development Environment*) itu merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk menggunakan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Bahasa pemrograman Arduino sendiri sekarang mengalami

perubahan sehingga mudah dipelajari oleh para pemula dalam melakukan pemrograman dari bahasa aslinya. Arduino IDE juga dilengkapi dengan *library* C/C++ yang biasa disebut wiring yang membuat operasi input dan output mejadi lebih mudah. Arduino IDE ini dikembangkan dari software processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino IDE.

## 2.5 Arduino Uno

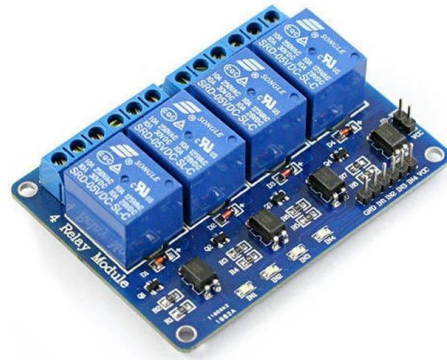
Arduino/Genuino Uno adalah papan mikrokontroler berbasis ATmega328P. ini memiliki 14 pin input/output digital (6 diantaranya dapat digunakan sebagai *output PWM*), 6 input analog, kristal kuarsa 16 MHz, konektor USB, konektor daya, header ICSP dan tombol reset. Ini berisi semua yang diperlukan untuk mendukung mikrokontroler, cukup sambung ke komputer dengan kabel USB sebagai sumber daya dengan adaptor AC - DC.



Gambar 2.4 Arduino Uno  
(Sumber: Arduinoindonesia, 2022)

## 2.6 Modul Relay

Relay adalah sakelar (*switch*) yang dioperasikan secara elektrik yang merupakan komponen elektromekanis yang terdiri dari dua bagian utama yakni elektromagnet (*coil*) dan bagian mekanis (kumpulan kontak saklar/*switch*). Relay menggunakan prinsip elektromagnetik untuk menggerakkan kontak sakelar sehingga dengan arus listrik yang kecil dapat menghantarkan arus yang bertegangan lebih tinggi. Misalnya, relay dengan elektromagnet 5VDC dan 50mA dapat menggerakkan Armature Relay (bertindak sebagai saklar) untuk mengalirkan arus 220V 2A (Lubis, 2019).



Gambar 2.6 Modul Relay 4 Channel  
(Sumber: Components101, 2021)



UNIVERSITAS  
**Dinamika**

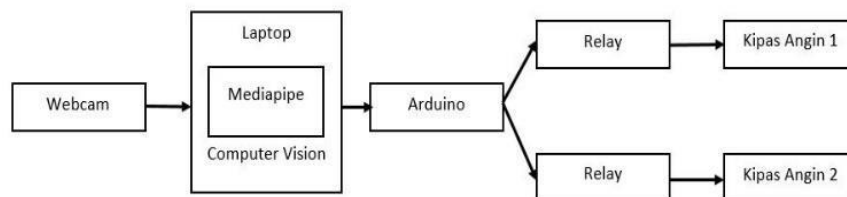
## BAB III METODOLOGI PENELITIAN

### 3.1 Instalasi *Environment*

Dalam penelitian ini, *environment* harus diinstall dan dikonfigurasi terlebih dahulu sehingga hal ini dilakukan agar *library* yang diperlukan untuk program dapat terinstall di program dengan baik dan dapat berjalan tanpa *crash*. Proses instal ini dibuat dengan menginstal Python 3.7.0, *library* dan *plugin* yang diperlukan di terminal pip. Arduino uno yang dikonfigurasi memiliki *String terminalRead* yang digunakan untuk membaca data dari terminal serial Arduino menggunakan bahasa *String*.

### 3.2 Perancangan Perangkat Keras

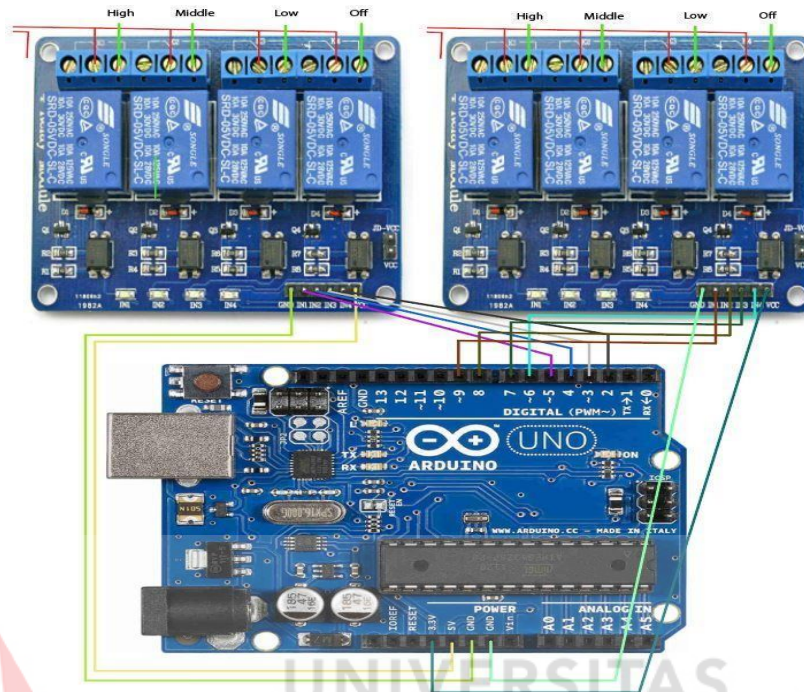
Struktur *hardware* dapat dilihat pada gambar dibawah ini dan inipunya yang berasal dari Webcam yang mengenali gesture jari tangan. Inputan tersebut kemudian diolah oleh PC menggunakan sistem *Computer Vision* dengan Mediapipe yang output tersebut akan ditampilkan di layar laptop. Setelah hasil output muncul, nilai Arduino akan menjalankan aksi pada relay untuk mengontrol atau mematikan kecepatan kipas angin dari hasil jari gestur tangan.



Gambar 3.1 Alur perancangan *Hardware*

Kemudian, pada diagram skematik Arduino yang dihubungkan ke relay. Lalu relay akan dihubungkan dengan tingkat putaran kipas angin, mulai dengan tingkat nol atau off, tingkat satu atau kecepatan (*low*), tingkat dua atau kecepatan sedang (*medium speed*) dan tingkat tiga atau kecepatan tinggi (*high*). NO yang artinya *Normal Open* dan NC yang artinya *Normal Close*, jadi pada saat keadaan off tidak ada tegangan, maka pin COM dan NO tidak terhubung, sedangkan pin

COM dengan CN terhubung. Tapi, ketika keadaan relay dihidupkan maka akan dikasih tegangan, lalu NO akan menjadi NC dan begitu sebaliknya, NC akan berubah menjadi NO. Ini yang sering digunakan sebagai sakelar dan kontak.



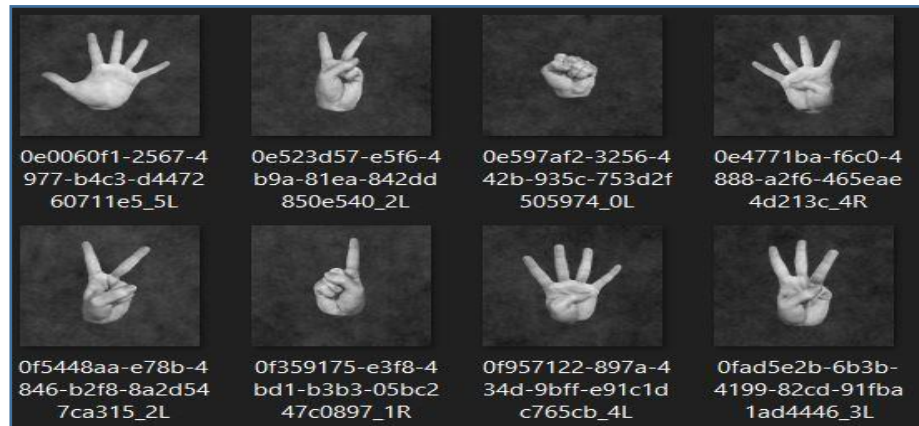
Gambar 3.2 Skema Hardware

Tabel 4.1 Tabel skema Hardware

No.	Arduino Uno	Relay 1 (4 Channel)	Relay 2 (4 Channel)
1.	Digital (PW) 02	IN4	-
2.	Digital (PW) 03	IN3	-
3.	Digital (PW) 04	IN2	-
4.	Digital (PW) 05	IN1	-
5.	Digital (PW) 06	-	IN4
6.	Digital (PW) 07	-	IN3
7.	Digital (PW) 08	-	IN2
8.	Digital (PW) 09	-	IN1
9.	Power GND	GND	GND
10.	Power VCC	VCC	VCC

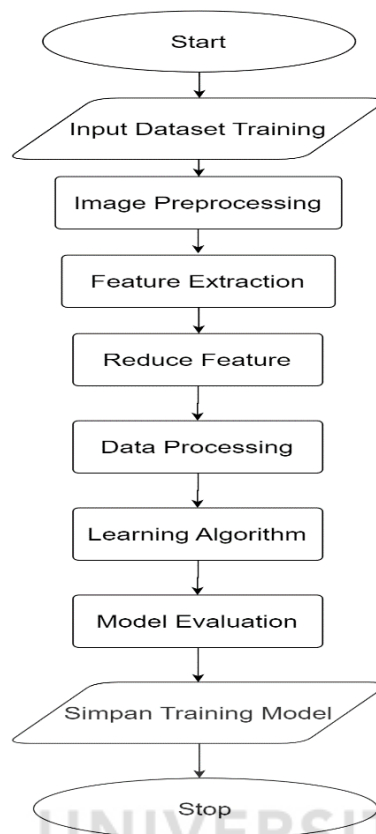


### 3.3 Dataset



Gambar 3.3 Dataset gestur jari tangan

Dataset akan digunakan dalam penelitian ini adalah dataset yang sudah dikumpulkan pada penelitian sebelumnya (Nautica, 2022). Dari penelitian sebelumnya ada 10 jari gestur tangan, namun yang digunakan penelitian ini menggunakan 6 gestur jari dan juga jari kepal dari kedua tangan kanan dan kiri. Dataset yang sudah dikumpulkan terdapat 21.600 gambar, sudah teraugmentasi dan juga mampu memenuhi kebutuhan penelitian. Pada aksi penambahan citra pada dataset dimodifikasi dan ditambahkan agar dataset lebih akurat dan beragam. Penulis menggunakan dataset ini untuk mengidentifikasi 6 gestur jari dari kedua tangan dan jari kepal. Keseluruhan dataset tersebut kemudian dibagi menjadi dua bagian dengan komposisi 80% untuk pelatihan dan 20% untuk validasi *Flowchart* untuk proses training dataset seperti gambar 3.5.



Gambar 3.4 *Flowchart* proses training dataset

Inisialisasi dari awal mulai dan lalu dataset pelatihan dimasukkan. Proses pra-pemrosesan citra kemudian mengubah intensitas piksel citra agar mudah digunakan pada proses selanjutnya. Kemudian ekstrak ciri-ciri untuk mengidentifikasi benda dengan ciri-ciri khusus dari benda tersebut. Lalu mengurangi properti untuk mendapatkan properti yang efektif dan menghilangkan properti yang tidak efektif. Setelah itu dilakukan pengolahan data untuk mengolah kumpulan data yang dipilih oleh Fitur. Kemudian dilakukan learning algoritma untuk mempelajari algoritma dari proses training dataset. Lalu Model Evaluation untuk mengevaluasi hasil data training akan disimpan dalam model. Terakhir menyimpan pola pelatihan ke file dalam format CKPT dan proses pelatihan selesai.

### 3.4 Metode Transmisi Serial

Dalam metode transmisi serial menggunakan string, data dikirimkan dalam format yang sesuai misalnya, jika ingin mengirimkan angka 8, dapat

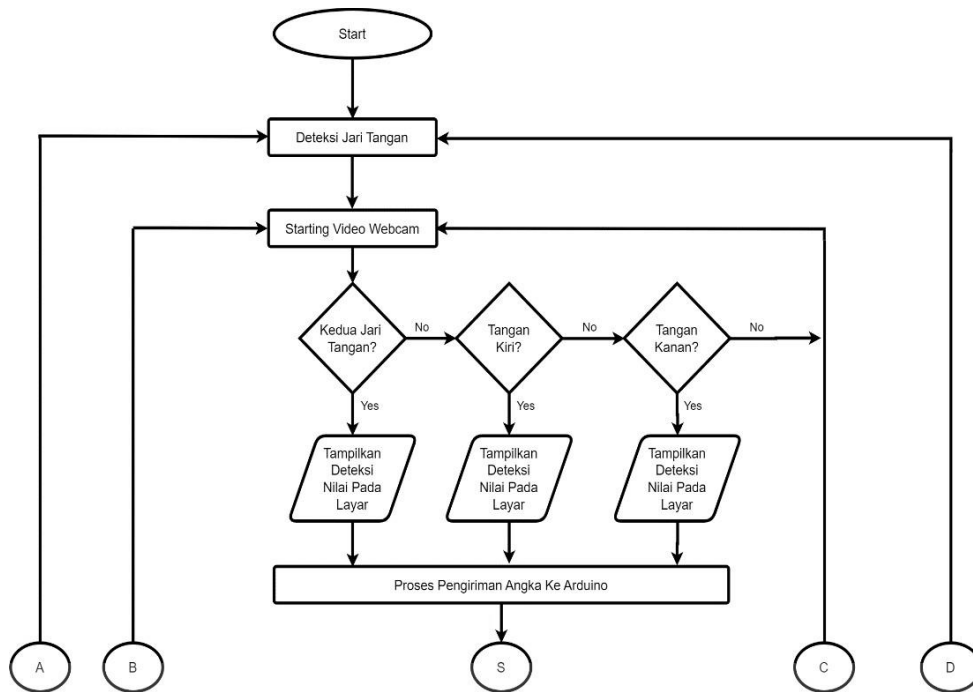
merepresentasikannya dalam format string sebagai 8. Data string ini dikirimkan melalui saluran komunikasi sesuai dengan protocol transmisi serial yang digunakan. Pada penelitian ini, terdapat 8 folder dalam format gambar yang sudah teraugmentasi yang mampu memenuhi kebutuhan penelitian. Proses ini juga merupakan proses yang dimana gambar pada dataset diubah serta ditambahkan jumlahnya sehingga dataset menjadi akurat dan bervariasi.

### **3.5 Metode Deteksi Dengan Mediapipe**

Pada metode Mediapipe, program mencari jari gesture tangan di Webcam/kamera. Saat jari gesture tangan terdeteksi, maka program menampilkan keypoint pada landmark yang dipakai untuk mengklasifikasi jari gesture tangan. Penerapan metode Mediapipe, setelah itu program mengenali jari gestur tangan dan melakukan ekstraksi landmark Mediapipe. Saat gesture jari tangan terdeteksi, sehingga program menampilkan jumlah jari tangan yang terdeteksi. Pada penelitian ini jarak yang diuji adalah 50cm, 100cm, 150cm, 200cm, 250cm. Habis itu, dicari rata-rata akurasi dan FPS dari setiap jari gesture tangan.

### **3.6 Proses *Testing* Program Python**

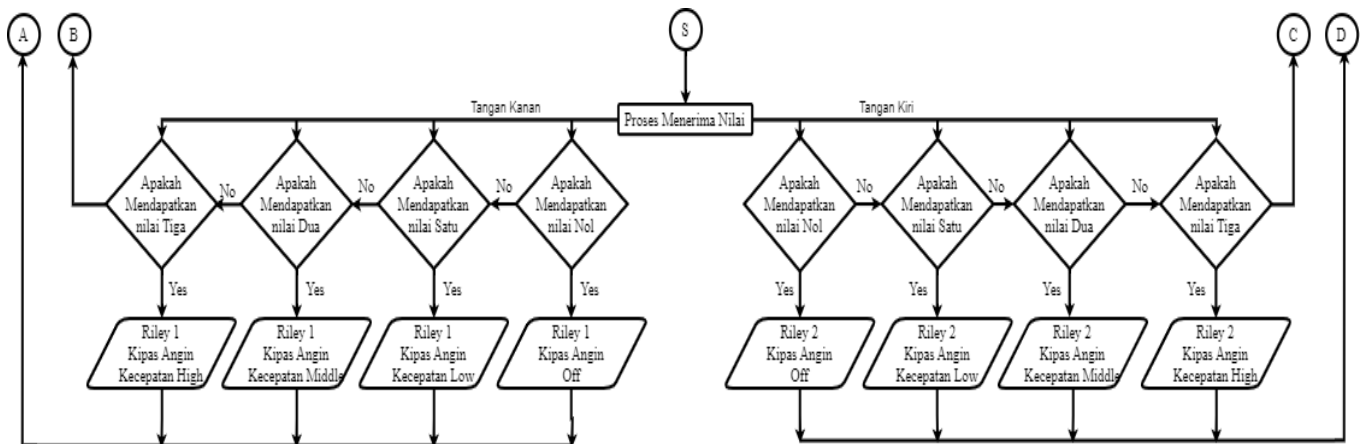
Dari gambar di bawah dapat dijelaskan bahwa input berasal dari Webcam yang mendeteksi gesture jari kedua tangan atau diantara kedua tangan yaitu mendeteksi gestur jari tangan nol, satu, dua, dan tiga. Setelah mendapatkan nilai yang 13 terdeteksi, kemudian tampilkan hasilnya di layar sebagai indikasi telah terdeteksi jari gestur tangan. Program kemudian mengirim hasil terdeteksi jari gesture tangan ke Arduino menggunakan komunikasi serial.



Gambar 3.5 Flowchart proses testing Program Python

### 3.7 Flowchart Program Mikrokontroler

Berdasarkan gambar dibawah ini, setelah dicek kondisi dimana nilai yang dihasilkan adalah angka 0,1,2 dan 3. Jika angka yang diterima 0 pada jari tangan kanan, relay 1 mematikan kipas angin. Jika nilai yang diterima adalah 1, relay 1 akan melakukan operasi kipas angin pada level pertama. Jika nilai yang dihasilkan adalah 2, relay 1 menjalankan kecepatan kipas angin pada *level* kedua. Jika nilai yang didapat 3, relay 1 menjalankan kecepatan kipas angin pada *level* ketiga. Jika nilai yang diterima 0 pada jari tangan kanan, relay 2 mematikan kipas angin. Jika nilai yang diterima adalah 1, relay 2 akan melakukan operasi kipas angin pada level pertama. Jika nilai yang dihasilkan adalah 2, relay 2 menjalankan kecepatan kipas angin pada *level* kedua. Jika nilai yang didapat 3, relay 2 melakukan aksi pada kecepatan kipas angin pada *level* ketiga. Kemudian kembali ke Webcam untuk mendeteksi jari gesture tangan.



Gambar 3.6 *Flowchart* proses testing program

### 3.8 *Computer Vision* dan Mikrokontroler Arduino menggunakan Komunikasi Serial

#### 3.8.1 Tujuan *Computer Vision* dan Mikrokontroler Arduino menggunakan Komunikasi Serial

Tujuan dibangunnya dari *Computer Vision* dan Mikrokontroler Arduino ini dengan komunikasi serial adalah untuk melihat bagaimana membangun koneksi serta komunikasi secara serial dari jalur *Computer Vision* Mediapipe ke mikrokontroler.

#### 3.8.2 *Computer Vision* dan Mikrokontroler Arduino Dengan Prosedur Membangun Komunikasi serial

Cara-cara untuk membuatnya adalah sebagai berikut:

1. Python dengan mengimport library *Pyserial*
2. Variabel komunikasi serial dideklarasikan
3. *Python* mengirimkan angka ke Arduino melalui koneksi serial
4. Arduino menerima angka dari koneksi serial untuk *python*

### 3.8.3 Hasil dari Membangun Komunikasi serial antara *Computer Vision* dan Mikrokontroler Arduino

Jika menggunakan fungsi serial dengan python, pertama menginstall Pustaka *pyserial cmd (Command Prompt)* atau *soft* yang digunakan, dengan mengetik:

```
C:\Users\Yumerius>pip install pyserial
```

Kemudian, mengimport *library pyserial* tersebut untuk program python dengan cara:

```
Import serial
```

“Import seial” Library yang berfungsi untuk transfer data antara program Python dan program mikrokontroler Arduino. Kemudian deklarasi sebuah variabel Bernama “*serialComm*” untuk mendapatkan port usb dan baud rate dari port serial USB di Arduino.

```
# Inisialisasi koneksi serial dengan Arduino
arduino = serial.Serial('COM5', 9600)
```

Kemudian kode ini mengirimkan nilai variabel *Serial* ke Arduino secara serial. Nilai akan dikirim ke Arduino bertipe data *String* variabel ini adalah variabel untuk kode tangan kanan, yang nilai string “1”.

```
data = str(count['RIGHT']) + ',' + str(count['LEFT']) + '\n'
arduino.write(data.encode())
```

Lalu kode ini mengirimkan nilai variabel counter ke Arduino secara serial. Angka yang dikirim ke Arduino tipe data *string*. Variabel ini adalah jumlah jari dari 0,1,2 dan 3. Kode berikut menyisipkan baris baru agar tidak tertukar dengan data *string*.

```
String terminalRead;
String terminaReadRight;
String terminalReadBothhand;
```

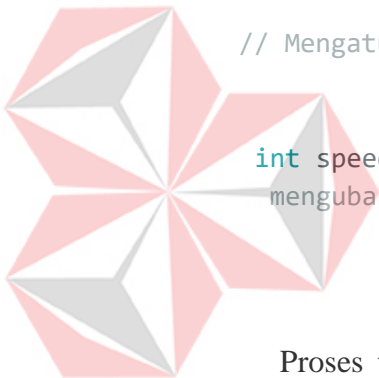
Pertama, mendeklarasikan program Arduino dengan variabel bernama *terminaRead* dengan tipe data *string*, kemudian variabel *terminalRead* diperintahkan untuk mendapatkan angka atau karakter dari hasil komunikasi serial

yang ada untuk mentransmisikannya ke data *string*. Oleh karena itu, variabel *terminalRead* merupakan hasil substring dari variabel string kanan dan kiri untuk mengaktifkan relay pada Arduino.

```
// memisahkan data untuk tangan kanan dan tangan kiri
terminalReadRight = receivedString.substring(0, 2);
terminalReadLeft = receivedString.substring(2, 4);
terminalReadBothHand = receivedString.substring(4, 6);

// Mengatur kecepatan kipas angin berdasarkan jumlah jari yang
// terdeteksi pada tangan kanan
if (terminalReadRight.length() > 0) {
int speed = map(terminalReadRight.toInt(), 0, 99, 0, 255); //
// mengubah kecepatan dari rentang 0-99 menjadi rentang 0-255
analogWrite(PORT_1, speed);
}

// Mengatur kecepatan kipas angin berdasarkan jumlah jari yang
// terdeteksi pada tangan kiri
if (terminalReadLeft.length() > 0) {
int speed = map(terminalReadLeft.toInt(), 0, 99, 0, 255); //
// mengubah kecepatan dari rentang 0-99 menjadi rentang 0-255
analogWrite(PORT_5, speed);
}
```



Proses terakhir adalah menyabungkan dari variabel baca terminal untuk mengatur kecepatan kipas dengan tangan kanan atau tangan kiri.

```
serial.serialutil.SerialException: could not open port 'COM5': FileNotFoundError(2, 'The system cannot find the file specified.', None, 2)
```

Gambar 3.7 Gagal dalam komunikasi serial

*Error* pada Gambar 3.7 adalah tidak bisa menyabungkan koneksi serial antara mikrokontroler dengan mediapipe (program python) *computer vision*.

```

INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
FPS : 1.0470585503958387
FPS : 43.5966613655971
FPS : 45.67117828326274
FPS : 27.849329712430364
FPS : 23.082024951434406
FPS : 32.40721653467259
FPS : 31.334683052556873
FPS : 21.790629773173595
FPS : 30.43982872487118
FPS : 45.41010122882044
FPS : 27.888401287268277
FPS : 33.02185551426592

```

Gambar 3.8 Python dapat mengeluarkan nilai

The screenshot shows the Arduino IDE interface with the following code in the sketch editor:

```

sketch_jun1a.ino
36 // memisahkan data untuk tangan kanan dan tangan kiri
37 terminalReadRight = receivedString.substring(0, 2);
38 terminalReadLeft = receivedString.substring(2, 4);
39 terminalReadBothHand = receivedString.substring(4, 6);
40
41 // Reset semua kipas angin ke posisi mati
42 for (int i = PORT_1; i <= PORT_8; i++) {
43   digitalWrite(i, HIGH);
44 }
45
46 // Mengatur kecepatan kipas angin berdasarkan jumlah jari yang terdeteksi pada tangan kanan
47 if (terminalReadRight.length() > 0) {
48   int speed = map(terminalReadRight.toInt(), 0, 99, 0, 255); // mengubah kecepatan dari rentang 0-99 menjadi rentang 0-255
49   analogWrite(PORT_1, speed);
50 }
51
52 // Mengatur kecepatan kipas angin berdasarkan jumlah jari yang terdeteksi pada tangan kiri
53 if (terminalReadLeft.length() > 0) {
54   int speed = map(terminalReadLeft.toInt(), 0, 99, 0, 255); // mengubah kecepatan dari rentang 0-99 menjadi rentang 0-255
55   analogWrite(PORT_5, speed);
56 }
57 }
58 }
59

```

The Output window displays the following information:

```

Sketch uses 4660 bytes (14%) of program storage space. Maximum is 32256 bytes.
Global variables use 216 bytes (10%) of dynamic memory, leaving 1832 bytes for local variables. Maximum is 2048 bytes.
Done uploading.

```

The status bar at the bottom indicates: Ln 45, Col 1 Arduino Uno on COM5

Gambar 3.9 Arduino dapat mengunggah program



Dari Gambar 3.8 dan 3.9, ketika Arduino IDE mampu menjalankan *coding*, yang terjadi adalah shell *Python* mencetak angka yang dicetak dalam *coding Python*. Dapat diartikan bahwa telah bisa berkomunikasi secara serial.



UNIVERSITAS  
Dinamika

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Bab iv mencakup hasil analisis dan pengujian, dimulai dengan perangkat lunak dengan perangkat keras. Selama fase pengujian perangkat lunak, memeriksa bahwa proses dataset dari pelatihan berjalan dengan baik. Kemudian dengan tahap pengujian perangkat keras memastikan bahwa alat yang digunakan bekerja sesuai yang diinginkan.

#### **4.1 Pengujian Metode Mediapipe**

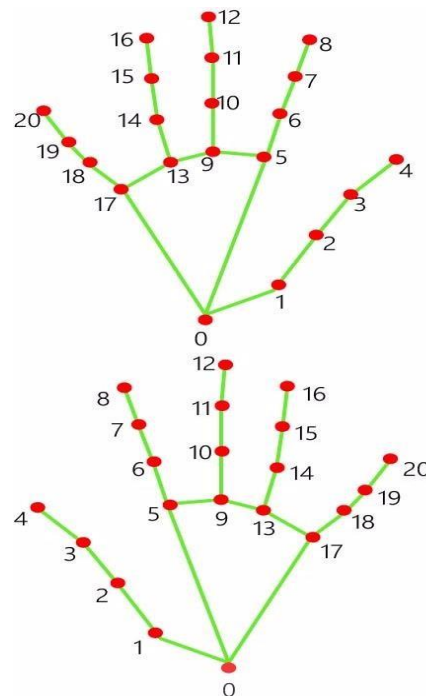
Dalam pengujian dengan menggunakan metode Mediapipe ini, diperlukan dataset untuk *training*, serta *library* Mediapipe untuk mengenali jari gestur tangan dengan *landmark* Mediapipe.

##### **4.1.1 Prosedur Pengujian Metode Mediapipe**

Cara-cara dalam melakukan pengujian ini adalah sebagai berikut:

1. Mengambil FPS dari jari gestur kedua tangan.
2. Mengambil FPS dan kecepatan kipas angin pada metode Mediapie.
3. Mengambil FPS dan jarak dari setiap gestur jari kedua tangan.

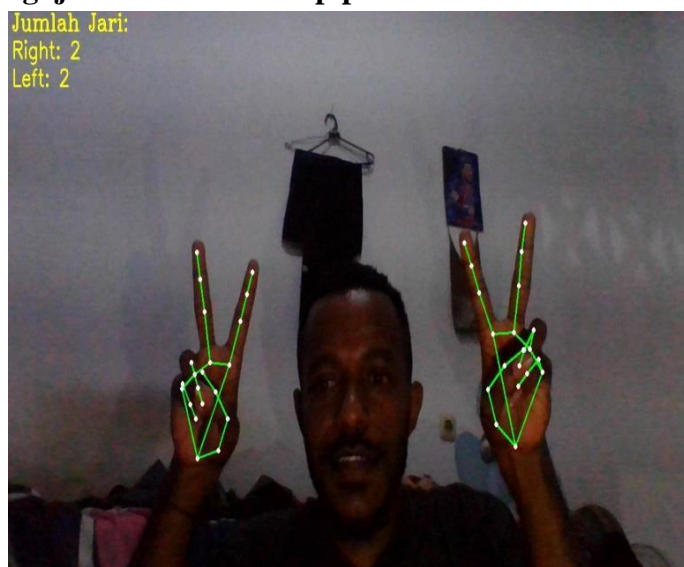
### 4.1.2 Keypoint Jari Kedua Tangan



- |                       |                       |
|-----------------------|-----------------------|
| 0. WRIST              | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC          | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP          | 13. RING_FINGER_MCP   |
| 3. THUMB_IP           | 14. RING_FINGER_PIP   |
| 4. THUMB_TIP          | 15. RING_FINGER_DIP   |
| 5. INDEX_FINGER_MCP   | 16. RING_FINGER_TIP   |
| 6. INDEX_FINGER_PIP   | 17. PINKY_MCP         |
| 7. INDEX_FINGER_DIP   | 18. PINKY_PIP         |
| 8. INDEX_FINGER_TIP   | 19. PINKY_DIP         |
| 9. MIDDLE_FINGER_MCP  | 20. PINKY_TIP         |
| 10. MIDDLE_FINGER_PIP |                       |

Gambar 4.1 Keypoint jari kedua tangan

### 4.1.3 Hasil Pengujian Metode Mediapipe



Gambar 4.2 Hasil deteksi gestur jari kedua tangan

Untuk Gambar 4.2, hasil dari uji gestur jari tangan menggunakan metode Mediapipe dapat mengenali gestur jari kanan 2 dan kiri 2. Deteksi jari gestur tangan ini memanfaatkan kamera handphone pada jarak 50cm dengan penerangan yang kurang merata. Gambar 4.2 menunjukkan bagaimana keypoint yang terbuka sehingga memberi nilai 2 untuk tangan kanan pada program karena 3 jari lainnya tidak terdeteksi dan nilai 2 untuk tangan kiri pada program karena 3 jari lainnya tidak terdeteksi setelah menghitung keypoint dari jari lainnya.

#### 4.2 Pengujian Jarak Metode Mediapipe

Dari Tabel 4.2 menguji nilai akurasi dan FPS dengan menguji program pengenalan sidik jari sebanyak 3 kali untuk setiap jarak. Pada subjek membuat empat gestur jari yang berbeda, dimulai dengan gestur angka nol, satu, dua, dan tiga, sehingga hasil deteksi jika benar 2 jari gestur tangan, maka bernilai 90% akurasi dan jika benar sekali nilainya 100% akurasi dengan melakukan 3 kali percobaan.

Tabel 4.2 menunjukkan akurasi pendeteksian dan *realime* dari semua jari gestur tangan. Metode Mediapipe yang diuji pada jarak 50cm sampai 250cm untuk tangan yang diuji yaitu jari kedua tangan. Pada penelitian ini dengan 1 subjek dan dilakukan 3 percobaan untuk setiap jari gestur tangan.

Tabel 4.2 Hasil uji jarak jari kedua tangan

No.	Jarak (cm)	Gestur Jari		Hasil Deteksi				Akurasi		FPS
		Kanan	Kiri	Benar		Salah		Kanan (%)	Kiri (%)	
				Kanan	Kiri	Kanan	Kiri			
1.	50	0	0	√	√	-	-	66,67	100	27.54
				-	√	√	-			34.23
				√	√	-	-			39.44
		1	1	√	√	-	-	100	100	29.39
				√	√	-	-			33.34
				√	√	-	-			33.23
		2	2	√	√	-	-	100	66,67	30.45
				√	√	-	-			39.67
				√	-	-	√			28.33
		3	3	√	√	-	-	100	100	32.34
				√	√	-	-			33.76
				√	√	-	-			35.00
2.	100	0	0	√	-	-	√	100	66,67	34.00
				√	√	-	-			31.82
				√	√	-	-			40.35
		1	1	√	√	-	-	66,67	100	35.65

		2	2	√	√	-	-	100	100	29.45		
				-	√	√	-			30.00		
				√	√	-	-			32.54		
				√	√	-	-			39.56		
				√	√	-	-			34.45		
				√	√	-	-			34.76		
		3	3	√	√	-	-	100	100	32.12		
				√	√	-	-			39.45		
				√	√	-	-			34.76		
				√	√	-	-			32.12		
				√	√	-	-			39.45		
				√	√	-	-			34.76		
3.	150	0	0	√	√	-	-	100	100	33.89		
				√	√	-	-			35.65		
				√	√	-	-			38.54		
		1	1	√	√	-	-	66,67	100	37.00		
				√	√	-	-			34.98		
				-	√	√	-			33.69		
		2	2	√	√	-	-	100	100	38.76		
				√	√	-	-			35.54		
				√	√	-	-			33.98		
		3	3	√	√	-	-	100	100	33.56		
				√	√	-	-			39.65		
				√	√	-	-			29.54		
		4.	200	0	0	√	√	-	-	66,67	100	25.56
						-	√	√	-			33.78
						√	√	-	-			37.34
				1	1	√	√	-	-	100	100	39.56
						√	√	-	-			34.58
						√	√	-	-			33.42
2	2			√	√	-	-	100	100	33.00		
				√	√	-	-			30.29		
				√	√	-	-			37.61		
3	3			√	√	-	-	100	100	33.43		
				√	√	-	-			34.89		
				√	√	-	-			33.90		
5.	250	0	0	√	√	-	-	100	66,67	33.21		
				√	-	-	√			33.43		
				√	-	-	√			34.87		
		1	1	√	√	-	-	100	100	39.00		
				√	√	-	-			35.87		
				√	√	-	-			33.98		
		2	2	√	√	-	-	100	100	36.65		
				√	√	-	-			32.51		
				√	√	-	-			33.32		
		3	3	√	-	-	√	66,67	66,67	29.67		
				-	√	√	-			34.21		
				√	√	-	-			32.90		
<b>Rata-rata Gestur Jari 0</b>								<b>86,67</b>	<b>86,67</b>	<b>34.36</b>		
<b>Rata-rata Gestur Jari 1</b>								<b>86,67</b>	<b>100</b>	<b>34.05</b>		
<b>Rata-rata Gestur Jari 2</b>								<b>100</b>	<b>93,33</b>	<b>33.15</b>		
<b>Rata-rata Gestur Jari 3</b>								<b>93,33</b>	<b>93,33</b>	<b>33.55</b>		

Pada Tabel 4.2 mencoba metode Mediapipe untuk setiap jarak, dengan rata-rata akurasi dan FPS untuk setiap jari gestur kedua tangan pada jarak yang berbeda. Hasil rata-rata nilai akurasi yang didapatkan pada setiap jarak sama dengan hasil rata-rata jarak 50cm sampai 250cm. Ini terjadi karena dari deteksi gestur pada jari yang paling dekat dengan kamera dan jauh dengan kamera, asalkan cahaya tersebar merata di setiap ruangan untuk mendeteksi gestur jari secara optimal.

### 4.3 Pengujian Deteksi Gestur Kedua Jari Tangan

Pada Tabel 4.3 dibawah ini, pengujian dari gestur kedua jari tangan FPS dengan mencoba program deteksi gestur jari kedua tangan yang diambil dengan sebanyak 3 kali untuk subjek. Setiap subjek yang melakukan pengujian diuji menggunakan 4 gestur jari tangan yang berbeda, dengan gestur angka 0, 1, 2 dan 3. Sehingga program mendeteksi 1 kali gestur tangan atau kedua jari tangan secara bersamaan, jika benar ditampilkan nilai FPS, jika hasil deteksi benar, nilainya 100.

Tabel uji 4.3 pengujian diuji dengan tangan kanan dan tangan kiri menggunakan metode Mediapipe. Subjek pada penelitian ini hanya 1 dengan 3 kali uji yang dilakukan untuk setiap jari gestur tangan dengan jarak yang digunakan 50cm.

Tabel 4.3 Hasil pengujian deteksi gestur jari kedua tangan

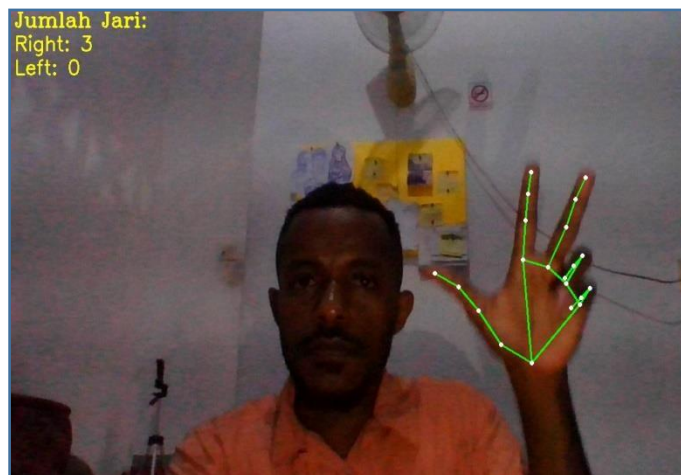
No.	Nama	Gestur Jari		Kipas Angin 1	Kipas Angin 2	Hasil Deteksi				FPS	Ket		
		Kanan	Kiri			Benar		Salah			Kanan	Kiri	
				Kanan	Kiri	Kanan	Kiri	Kanan	Kiri				
1.	Yume	0	0	√	√	√		-	√	35.20	-	Output salah deteksi sehingga sistem berpindah ke jari 1	
				√	√	√	√	-	-	30.44	-	-	
				√	√	√	√	-	-	36.53	-	-	
		1	1	√	√	√	√	-	-	37.45	-	-	
				√	√	√	√	-	-	33.37	-	-	
				√	√	√	√	-	-	30.78	-	-	
		2	2	√	√	√	√	√	-	-	29.59	Output salah deteksi sehingga sistem berpindah ke jari 3	-
				√	√	√	√	-	-	33.34	-	-	
				√	√	√	√	-	-	35.98	-	-	
		3	3	√	√	√	√	-	-	34.00	-	-	
				√	√	√	√	-	-	32.54	-	-	
				√	√	√	√	-	-	37.54	-	-	
2.	Vausan	0	0	√	√	√	√	-	-	33.19	-	-	
				√	√	√	√	-	-	34.60	-	-	
				√	√	√	√	-	-	35.57	-	-	
		1	1	√	√	√	√	-	-	29.34	-	-	
				√	√	√	√	-	-	34.31	-	-	

No.	Nama	Gestur Jari		Kipas Angin 1	Kipas Angin 2	Hasil Deteksi				FPS	Ket				
		Kanan	Kiri			Benar		Salah			Kanan	Kiri			
				Kanan	Kiri	Kanan	Kiri	Kanan	Kiri						
3.	Ilham	0	0	√	√	√	√	-	-	28.45	-	-			
				√	√	√	√	-	-	37.54	-	-			
		2	2	√	√		√	√	-	-	32.45	Output salah deteksi sehingga sistem berpindah ke jari 3	-		
				√	√	√	√	-	-	33.90	-	-			
		3	3	√	√	√	√	-	-	37.61	-	-			
				√	√	√	√	-	-	32.67	-	-			
				√	√	√	√	-	-	31.00	-	-			
		4.	Niol	0	0	√	√		√	√	-	-	39.12	Output salah deteksi sehingga sistem berpindah ke jari 1	-
						√	√	√	√	-	-	32.12	-	-	
				1	1	√	√	√	√	-	-	40.10	-	-	
						√	√	√	√	-	-	32.34	-	-	
				2	2	√	√	√	√	-	-	38.45	-	-	
√	√					√	√	-	-	35.54	-	-			
3	3			√	√	√	√	-	-	32.40	-	-			
				√	√	√	√	-	-	33.90	-	-			
				√	√	√	√	-	-	31.78	-	-			
				√	√	√	√	-	-	36.32	-	-			
				√	√	√	√	-	-	37.09	-	-			
				√	√	√	√	-	-	32.34	-	-			
		√	√	√	√	-	-	35.43	-	-					
		√	√	√	√	-	-	38.12	-	-					
		√	√	√	√	-	-	32.10	-	-					
5.	Yose	0	0	√	√	√	√	-	-	35.58	-	-			
				√	√	√	√	-	-	33.33	-	-			
		1	1	√	√	√	√	-	-	34.65	-	-			
				√	√	√	√	-	-	31.99	-	-			
		2	2	√	√	√	√	-	-	32.13	-	-			
				√	√	√	√	-	√	33.19	-	Output salah deteksi sehingga sistem berpindah ke jari 3			
		3	3	√	√	√	√	-	-	34.67	-	-			
				√	√	√	√	-	-	40.54	-	-			
				√	√	√	√	-	-	39.55	-	-			
				√	√	√	√	-	-	30.34	-	-			
				√	√	√	√	-	-	34.45	-	-			
				√	√	√	√	-	-	32.11	-	-			
		√	√	√	√	-	-	30.34	-	-					
		√	√	√	√	-	-	30.00	-	-					
		√	√	√	√	-	-	38.67	-	-					
		√	√	√	√	-	-	32.61	-	-					
		√	√	√	√	-	-	33.98	-	-					
		√	√	√	√	-	-	37.88	-	-					
		√	√	√	√	-	-	31.82	-	-					
		√	√	√	√	-	√	32.56	-	Output salah deteksi sehingga sistem berpindah ke jari 3					
		√	√	√	√	-	-	33.22	-	-					

Keypoints saat mendeteksi gestur jari 1, deteksi yang salah menjadi gestur 2, yang menghasilkan perbedaan FPS yang tidak seimbang untuk FPS gestur jari 0.

Dari hasil FPS diambil keluaran *realtime* saat program berjalan, lalu diambil nilai FPS dari setiap subjek untuk seriap percobaan.

#### 4.4 Pengujian Kontrol Kecepatan Kipas Angin



Gambar 4.3 Hasil deteksi gestur jari untuk mengontrol kecepatan kipas

Dari Gambar 4.4 pada hasil pengenalan gestur jari untuk menghidupkan dan mengontrol kipas angin telah berjalan sesuai dengan harapan peneliti. Subjek melakukan 3 gestur jari tangan kanan, relay menjalankan aksi untuk menghidupkan kipas angin dengan fungsi kecepatan kipas sesuai dengan salah data yang diterima dari Arduino.

Tabel pengujian 4.4 diuji dengan menggunakan jari kedua tangan kanan dan kiri dengan menggunakan metode Mediapipe. Pada penelitian ini dilakukan 3 kali percobaan dengan 1 subjek subjek untuk setiap gestur jari tangan dengan jarak yang digunakan 50cm. Pada subjek membuat empat jari gestur yang berbeda, dengan gestur jari nol, satu, dua, dan tiga.

Tabel 4.4 Hasil pengujian kontrol kecepatan kipas angin

No.	Nama	Gestur Jari		Output		Kecepatan Kipas Angin								Hasil				FPS	K t			
		Kanan	Kiri	Kanan	Kiri	0		1		2		3		Benar		Salah			Ka	Ki		
						Ka	Ki	Ka	Ki	Ka	Ki	Ka	Ki	Ka	Ki	Ka	Ki					
1.	Yum	0	0	√	√	√	√	-	-	-	-	-	-	√	√	-	-	33.24	-	-		
				√	√	√	√	-	-	-	-	-	-	√	√	-	-	36.21	-	-		
				√	√	√	√	-	-	-	-	-	-	-	√	-	√	-	√	39.23	-	sistem berpindah ke jari 1
				√	√	-	-	√	√	-	-	-	-	-	√	√	-	-	33.20	-	-	
				√	√	-	-	√	√	-	-	-	-	-	√	√	-	-	35.32	-	-	
				√	√	-	-	√	√	-	-	-	-	-	√	√	-	-	31.45	-	-	
		1	1	√	√	-	-	-	-	√	√	-	-	-	√	√	-	-	33.49	-	-	
				√	√	-	-	-	-	√	√	-	-	-	√	√	-	-	33.00	-	-	

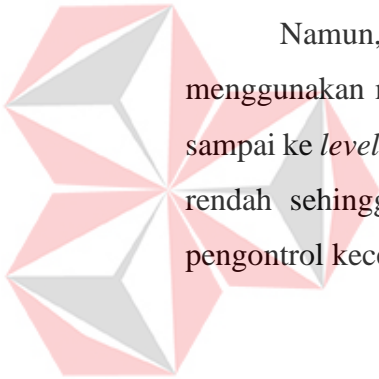


No.	Nama	Gestur Jari		Ouuput		Kecepatan Kipas Angin												H il				FPS	K t					
		Kanan	Kiri	Kanan	Kiri	0				1				2				3					Benar		Salah		Ka	Ki
						Ka	Ki	Ka	Ki	Ka	Ki	Ka	Ki	Ka	Ki	Ka	Ki	Ka	Ki	Ka	Ki							
2.	Ham	0	0	√	√	-	-	-	-	√	√	-	-	√	√	-	-	√	√	-	-	31.22	-	-	sistem berpindah ke jari 1			
				√	√	-	-	-	-	-	-	-	-	√	√	-	-	√	√	-	-	35.65	-	-				
				√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	√	√	-	-	38.54		-	-	
				√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	√	√	-	-	39.77		-	-	
		1	1	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	26.45		-	-	
				√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	35.76		-	-	
				√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	32.65		-	-	
				√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	31.23		-	-	
				√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	29.43		-	-	
				√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	30.59		-	-	
				√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	27.43		-	-	
				√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	39.45		-	-	
2	2	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	34.55	-	-				
		√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	32.67	-	-				
		√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	29.65	-	-				
		√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	40.54	-	-				
3	3	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	33.45	-	-				
		√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	30.00	-	-				
		√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	29.56	-	-				
		√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	37.49	-	-				
		√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	32.98	-	-				
		√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	33.98	-	-				
		√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	28.45	-	-				
		√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	39.93	-	-				
3.	Fred	0	0	√	√	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	30.00	-	-	sistem berpindah ke jari 2			
				√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	29.56	-		-		
				√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	37.49		-	-	
				√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	32.98		-	-	
		1	1	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	33.98		-	-	
				√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	28.45		-	-	
				√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	39.93		-	-	
				√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	34.51		-	-	
		2	2	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	33.44		-	-	
				√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	39.21		-	-	
				√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	29.54		-	-	
				√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	35.85		-	-	
0	0	√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	38.43	-	-				
		√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	33.56	-	-				
		√	√	-	-	-	-	√	√	-	-	-	-	-	-	-	-	√	√	√	-	33.89	-	-				
		√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	34.51	-	-				
		√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	33.66	-	-				
		√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	34.56	-	-				
		√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	33.78	-	-				
		√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	37.54	-	-				
1	1	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	√	-	31.43	-	-				
		√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	29.45	-	-				
		√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	30.54	-	-				
		√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	36.56	-	-				
4.	Yos	0	0	√	√	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	30.21	-	-				
				√	√	√	√	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	34.76	-	-		
				√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	√	-	33.24	-	-		
		1	1	√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	34.78	-	-		
				√	√	-	-	√	√	-	-	-	-	-	-	-	-	-	-	√	√	-	-	33.95	-	-		
				√	√	-	-	-	-	√	√	-	-	-	-	-	-	-	-	√	√	-	-	38.79	-	-		
				√	√	-	-	-	-	√	√	-	-	-	-	-	-	-	-	√	√	-	-	32.22	-	-		
		2	2	√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	30.23	-	-		
				√	√	-	-	-	-	-	-	√	√	-	-	-	-	-	-	√	√	-	-	29.00	-	-		
				√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	37.65	-	-		
				√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	33.95	-	-		
		3	3	√	√	-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	33.30	-	-		
√	√			-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	33.30	-	-				
√	√			-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	33.30	-	-				
√	√			-	-	-	-	-	-	-	-	-	-	-	-	-	-	√	√	-	-	33.30	-	-				
<b>Jumlah benar dan salah dari kedua jari tangan pada pengujian kontrol putaran kecepatan kipas angin</b>														58	60	4	2	33.30	-	-								

Dalam hasil pengujian pada Tabel 4.4 bahwa Ka dan Ki merupakan singkatan dari kedua tangan Kanan dan tangan Kiri. Tabel 4.4 memperlihatkan hasil *finger gesture recognition* untuk mengatur kecepatan putaran dua buah kipas setelah dilakukan pengujian dengan sebuah *recognition* dengan 8 *finger gesture* yang berbeda untuk kedua tangan, dimulai dari angka 0, 1, 2 dan 3. Dari tabel tersebut diperoleh hasil harus cocok dengan gerakanyang terdeteksi oleh jari. Jika salah, juga akan mempengaruhi sistem pengatur kecepatan kedua kipas. Sistem kontrol kecepatan kipas terhubung ke sistem pengenalan sidik jari melalui koneksi serial antara mikrokontroler Arduino dan *Computer Vision*.

Jari tangan kanan benar bernilai 85% dan salah 15%, untuk jari tangan kiri jika benar bernilai 90% dan kesalah 10%. Jika benar sekali, maka kedua jari tangan bernilainya 100% dan kesalah 0%.

Namun, sistem ini dapat berjalan sebagai pengontrol kecepatan kipas angin menggunakan metode Mediapipe. Sistem pendeteksi gestur jari tangan juga tidak sampai ke *level* ideal, karena spesifikasi dari perangkat keras yang digunakan masih rendah sehingga FPS turun dan program menjadi tidak responsive, sehingga pengontrol kecepatan kipas juga tidak berfungsi karena deteksi gerakan jari salah.



UNIVERSITAS  
Dinamika

## BAB V PENUTUP

### 5.1 Kesimpulan

Dari menguji sistem bentuk jari gestur kedua tangan hingga mengontrol kecepatan putaran dua kipas angin hingga pengujian jarak yang dirancang pada penelitian ini, maka ada kesimpulan yang dapat diusulkan:

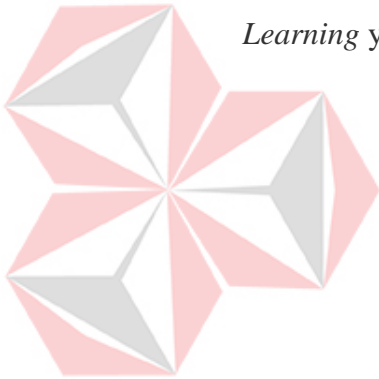
1. *Computer Vision* dan mikrokontroler Arduino saat membuat koneksi serial tes ini berhasil menghubungkan *computer vision* dengan mikrokontroler Arduino sehingga tes ini dapat mengenali kedua jari sekaligus secara *Automatic Feature Selection*.
2. Hasil pengujian untuk mendeteksi kedua jari tangan yang telah diuji, bahwa hasil yang didapatkan dari kedua jari tangan kanan dan kiri dengan kipas angin 1 dan kipas angin 2 bahwa telah berjalan dengan baik, namun terkadang sistem salah deteksi sehingga dapat berpindah gestur jari.
3. Untuk mengecek putaran pada kecepatan kedua kipas angin saat menguji sistem pengenalan gestur jari tangan, dilakukan 3 kali percobaan dengan menggunakan metode Mediapipe untuk mengontrol kipas angin. Pengujian ini dilakukan dengan jarak 50cm. Hasil yang jari tangan kanan yang benar dengan nilai 85% (58 benar), dapat dikatakan pengenalan sidik jari bekerja dengan baik, namun hasilnya salah dengan nilai 15% (4 kesalahan), pada jari tangan kiri yang benar dengan nilai 90% (60 benar), ini dapat dikatakan bahwa pengenalan pada jari berjalan baik, namun hasil yang salah dengan nilai 10% (2 kesalahan) hal ini dikarenakan kesalahan dalam pengenalan gestur jari tangan yang belum mencapai tingkat yang sempurna, pada hasil percobaan ini juga mendapatkan hasil rata-rata FPS 33.30.
4. Hasil rata-rata nilai akurasi dan *error* semua deteksi gestur jari yang dilakukan dengan metode Mediapipe dengan jarak 50cm sampai 250cm, dilakukan 3 kali percobaan pada setiap jarak, pada tangan kanan tingkat akurasi gestur jari 0 sebesar 86,67%, akurasi gestur 1 sebesar 86,67%, akurasi gestur 2 sebesar 100%, akurasi gestur 3 sebesar 93,33 % dan pada tangan kiri tingkat akurasi gestur jari 0 sebesar 86,67%, akurasi gestur 1 sebesar 100%, akurasi gestur 2

sebesar 93,33%, dan akurasi gestur 3 sebesar 93,33%. Pada percobaan ini juga menghasilkan nilai rata-rata FPS untuk setiap gestur jari tangan dengan rata-rata gestur jari 0 sebesar 34.36, gestur jari 1 sebesar 34.05, gestur jari 2 sebesar 33.15 dan gestur jari 3 sebesar 33.55.

### 5.1 Saran

Sehubungan dengan saran untuk pengembangan lebih lanjut penelitian ini untuk memperbaikinya, maka saran yang diberikan adalah sebagai berikut:

1. Saat menggunakan metode untuk mendeteksi bentuk gestur yang lebih efisien, di sarankan menggunakan arsitektur *YOLO* dan *SSD*
2. Dapat dikembangkan dengan mengimplementasikan sistem deteksi gestur jari tangan menggunakan objek yang lain.
3. Adapun metode pada penelitian ini bisa dikembangkan menggunakan *deep Learning* yang berjalan secara *realtime* contohnya *SSD*, *YOLO* dan *LSTM*



UNIVERSITAS  
**Dinamika**

## DAFTAR PUSTAKA

- Andrian Rosebrock, 2018. *internet*.  
<https://pyimagesearch.com/2018/07/19/opencv-tutorial-a-guide-to-learn-opencv/>
- Arduino Indonesia 2022. *Internet*.  
<https://www.arduinoindonesia.id/2022/08/pengertian-dan-penjelasan-arduino-uno.html>
- Arisandi, E. D. (2014). Kemudahan Pemrograman Mikrokontroler Arduino Pada Aplikasi Wahana Terang. SETRUM, 46-49.
- Component101 2021. *Internet*. <https://components101.com/switches/5v-four-channel-relay-module-pinout-features-applications-working-datasheet>
- Fakhrudin, Muhammad Aldi (2023). Sistem Deteksi Gestur Jari Tangan menggunakan Mediapipe dan Faster-RCNN untuk Mengontrol Kecepatan Kipas Angin. Repository Univesitas Dinamika.
- Logos-World. *Internet*. <https://logos-world.net/python-logo/>
- Lubis, Z. (2019). Kontrol Mesin Air Otomatis Berbasis Arduino Dengan Smartphone. Buletin Utama Teknik, 155-159.
- Mediapipe. (2019). Live Machine Learning Anywhere. Von Mediapipe Developer. <https://mediapipe.dev/abgerufen>.
- Mediapipe Hand 2006. *Internet*. <https://www.arxiv-vanity.com/papers/2006.10214/>
- Nautica, M. R. (2022). Hand Gesture Detection sebagai Alat Bantu Ajar Berhitung menggunakan Mediapipe dan Convolutional Neural Network secara Realtime. Repository Universitas Dinamika, 1-44.
- Ramdhon, A. N. (2021). Penerapan Face Recognition Pada Sistem Presensi.JACOST, 12-17.
- Zein, A. (2018). Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON. Sainstech, 22-26.