

---

## KONTROL KIPAS ANGIN SECARA JARAK JAUH MELALUI PENGENALAN BENTUK GESTUR JARI TANGAN BERBASIS *COMPUTER VISION*

Oleh

Fredi Wakerkwa<sup>1</sup>, Heri Pratikno<sup>2\*</sup>, Weny Indah Kusumawati<sup>3</sup>, Musayyanah<sup>4</sup>

<sup>1,2,3,4</sup> Program Studi Teknik Komputer, Universitas Dinamika

E-mail: <sup>1</sup>[edywaker99@gmail.com](mailto:edywaker99@gmail.com), <sup>2\*</sup>[heri@dinamika.ac.id](mailto:heri@dinamika.ac.id), <sup>3</sup>[weny@dinamika.ac.id](mailto:weny@dinamika.ac.id),  
<sup>4</sup>[musayyanah@dinamika.ac.id](mailto:musayyanah@dinamika.ac.id)

---

### **Article History:**

Received: 21-07-2023

Revised: 30-07-2023

Accepted: 24-08-2023

### **Keywords:**

Gestur jari, MediaPipe, MQTT, Visi Komputer, WeMos

**Abstract:** Artificial Intelligence berkembang sangat pesat, visi komputer bagian dari kecerdasan buatan untuk memproses data visual. Disisi lain mikrokontroler juga berkembang dengan pesat guna memudahkan tugas manusia. Pada penelitian ini melakukan pengontrolan kipas angin secara jarak jauh melalui proses deteksi gestur jari tangan menggunakan metode Mediapipe serta modul WeMos berbasis protokol MQTT. Hasil percobaan dari penelitian ini mendapatkan akurasi deteksi gestur tangan mematikan kipas (0, tangan mengepal) pada jarak 100 cm sebesar 100% dengan 28,6 FPS, akurasi deteksi gestur jari tangan untuk menghidupkan kipas angin pada level kecepatan 1 sebesar 94% pada 29,9 FPS, sedangkan akurasi deteksi gestur 2 untuk mengontrol kecepatan putaran kipas angin pada level 2 adalah 99% pada 28,6 FPS, dan hasil deteksi gestur jari tangan untuk mengendalikan kecepatan putaran kipas angin pada level 3 mencapai 99. % akurasi pada 31,4 FPS. Pengujian pada jarak 150 cm menunjukkan akurasi 100% untuk deteksi isyarat 0, 1, 2, dan 3 pada lima orang. Pada jarak 200 cm akurasi deteksi gestur 0 dan 1 sebesar 80%, sedangkan gestur 2 dan 3 mempertahankan akurasi 100%. Di atas 200 cm, gestur 0 dan 1 mencapai akurasi 80%, gestur 2 mencapai akurasi 100%, dan gestur 3 mencapai akurasi 90%.

---

## PENDAHULUAN

Terjadi perkembangan pesat teknologi dalam bidang IoT (*Internet of Things*), AI (*Artificial Intelligence*), dan *Computer Vision* (CV). IoT memungkinkan fungsi kontrol, remote, monitoring peralatan elektronik secara jarak jauh oleh pengguna darimana saja dan kapan saja. Pada AI (*Artificial Intelligence*) dan CV (*Computer Vision*) mampu membuat mesin atau komputer untuk memahami dan memproses informasi dari sebuah gambar maupun video. Penerapan teknologi AI, visi komputer serta IoT dapat dilakukan dalam berbagai sektor,

diantaranya: pertanian, manufaktur, transportasi, dan medis secara otomatis, sehingga dapat mempermudah pengawasan dan kontrol pada sistem yang kompleks. Visi komputer merupakan bidang studi tentang bagaimana komputer dapat mengambil, memproses, serta mempelajari informasi visual pada dunia nyata yang melibatkan penggunaan algoritma dan teknologi untuk mengubah citra atau video menjadi data yang dapat dipahami serta dimanfaatkan oleh komputer untuk berbagai aplikasi, diantaranya: pengenalan wajah, deteksi objek, analisis citra medis, dan robotika (Szeliski, 2022). Penerapan visi komputer untuk mendeteksi gestur jari tangan menggunakan MediaPipe juga dapat digunakan untuk mengontrol pergerakan *mouse* (Hernando, 2023).

Mikrokontroler sebagai *backbone* penggerak dalam lingkungan IoT dapat diterapkan dalam kehidupan sehari-hari untuk fungsi kontroling, monitoring, maupun pengamanan dengan tujuan untuk mempermudah pekerjaan dan meningkatkan kualitas hidup manusia. Hasil dari penelitian sebelumnya yang dilakukan Muhammad Aldi Fakhruddin pada tahun 2023 yang berjudul Sistem Deteksi Gestur Jari Tangan menggunakan Mediapipe dan Faster-RCNN untuk Mengontrol Kecepatan Kipas Angin (Fakhruddin, 2023) menjadi landasan utama bagi peneliti untuk dikembangkan lebih lanjut dengan kontribusi penerapan konsep proses deteksi bentuk gestur jari tangan untuk mengontrol level putaran kecepatan kipas angin secara jarak jauh. Penelitian lainnya dengan judul *Optimized Hand Gesture Based Home Automation for Feebles* (R, 2022), bertujuan mengembangkan sebuah sistem pengaturan perangkat di rumah yang dapat dikendalikan melalui gerakan tangan dan pola penggunaannya. Penelitian lain yang mengontrol kipas angin dan lampu menggunakan saklar suara berbasis Arduino Uno (Fifit Alfiah, Budi Rahman and Imelda, 2020).

Pada penelitian ini, penulis melakukan proses pengontrolan untuk mematikan (OFF) serta menghidupkan (ON) kipas angin berdasarkan level kecepatan putarannya (*level 1, 2 dan 3*) secara visi komputer melalui deteksi bentuk gestur jari tangan secara otomatis dari jarak jauh (internet) berbasis protokol komunikasi jaringan MQTT menggunakan modul mikrokontroler WeMos-D1R2 (ESP8266). Pengontrolan kipas angin tersebut dilakukan secara visi komputer menggunakan MediaPipe tanpa perlu sensor apapun hanya menggunakan sebuah *webcam* dari *laptop* sehingga proses interaksinya lebih alami, natural dan intuitif karena tidak diperlukan lagi kontak fisik dari pengguna, proses pengontrolan tersebut dapat dilakukan oleh jari tangan kanan maupun jari tangan kiri. Keluaran dari proses komputasi secara visi komputer tersebut selanjutnya ditransmisikan secara jarak jauh menggunakan protokol jaringan MQTT, sehingga keluaran dari penelitian ini sudah menerapkan teknologi yang berbasis *Internet of Things* (IoT).

## LANDASAN TEORI

Dalam menyelesaikan masalahnya kecerdasan manusia dipengaruhi oleh pengetahuan dan pengalamannya. Pengetahuan diperoleh melalui proses belajar, kemampuan berpikir secara rasional, menggunakan pengetahuan serta pengalaman yang dimilikinya dapat digunakan untuk pengambilan keputusan yang lebih tepat. Pengetahuan dan pengalaman tidak ada gunanya dalam menyelesaikan masalah jika tidak mempunyai kemampuan berpikir yang baik, Sebaliknya, kemampuan berpikir manusia tidak mencukupi untuk menyelesaikan masalah dengan baik tanpa pengetahuan dan pengalaman yang memadai. Untuk bisa bertindak seperti manusia, mesin harus diberi pengetahuan melalui proses

*training* yang cukup dan kemampuan berpikir yang baik (Sobron and Lubis, 2021).

Bahasa pemrograman Python sejak kemunculannya pada tahun 1990-an menjadi opsi bahasa pemrograman yang paling populer di industri. Berdasarkan survei yang dilakukan oleh RedMonk pada tahun 2021, menempatkan bahasa Python menjadi peringkat kedua sebagai bahasa pemrograman paling favorit para pengembang sekitar 30 tahun sejak awal diluncurkan (Lugaresi *et al.*, 2019). MediaPipe adalah sebuah *platform* yang digunakan untuk pembelajaran mesin yang dapat memproses data *time-series* seperti *video*, *audio*, dan sebagainya. MediaPipe dapat dijalankan pada beberapa *platform* seperti *Desktop/Server*, iOS, Android, serta perangkat lainnya seperti Raspberry Pi dan Jetson Nano dengan bentuk tampilan programnya tampak pada Gambar 1. WeMos-D1R1 merupakan papan pengembangan yang kompatibel dengan Arduino Uno yang sudah memiliki modul Wifi ESP8266. WeMos D1R2 berbasiskan pada chip ESP8266 yang dapat terhubung ke jaringan WiFi sebagaimana terlihat pada Gambar 2. Penelitian yang menerapkan *computer vision* dan *deep learning* menggunakan *framework* MediaPipe dan arsitektur CNN sebagai alat bantu ajar berhitung pada pendidikan anak usia dini melalui deteksi gestur kesepuluh jari tangan secara *realtime* (Nautica, 2022).



**Gambar 1. Tampilan keluaran program Mediapipe  
(Sumber: (Kukil, 2022))**

Arduino IDE merupakan program yang digunakan membuat dan mengedit kode pemrograman atau sketch pada *board* Arduino. Arduino IDE memungkinkan pengguna dengan mudah untuk menulis, mengunggah, dan memprogram *board* Arduino menggunakan bahasa pemrograman C/C++ (Suprianto *et al.*, 2019). IoT (*Internet of Things*) adalah sistem *embedded* yang bertujuan untuk memperluas pemanfaatan dari konektivitas internet yang tersambung secara terus-menerus (Susanto, Prasiani and Darmawan, 2022). Perangkat IoT bisa mengumpulkan data tentang kondisi lingkungan sekitarnya, seperti: suhu, kelembaban, lokasi, dan sebagainya yang dapat digunakan untuk mengoptimalkan penggunaan sumber daya, meningkatkan efisiensi, dan menghadirkan pengalaman yang lebih personal.

MQTT (*Message Queuing Telemetry Transport*) adalah protokol yang beroperasi menggunakan konsep TCP/IP yang prinsip kerjanya serupa dengan model *client-server*, pada MQTT dikenal dengan istilah *publisher-subscriber* (Fifit Alfiah, Budi Rahman and Imelda, 2020). MQTT banyak digunakan untuk komunikasi antar-perangkat di *Internet of Things* (IoT) dengan ukurannya yang kecil dan efisien. MQTT mendukung koneksi ke berbagai

platform dan bahasa pemrograman, *publisher* bertanggung jawab mengirimkan pesan ke *broker*, adapun *subscriber* untuk berlangganan pesan dari *broker*.

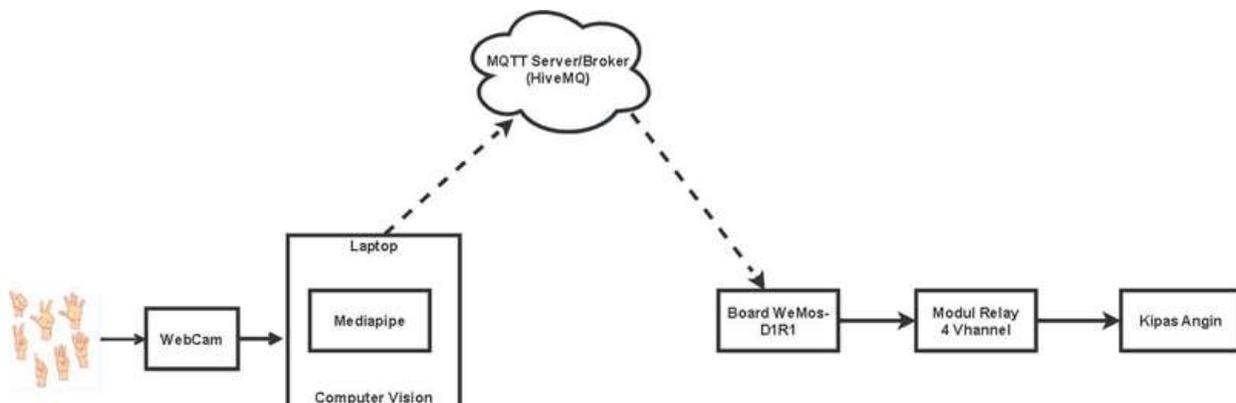


Gambar 2. Board WeMos-D1R2 yang berbasis ESP8266  
(Sumber: (Andy, 2019))

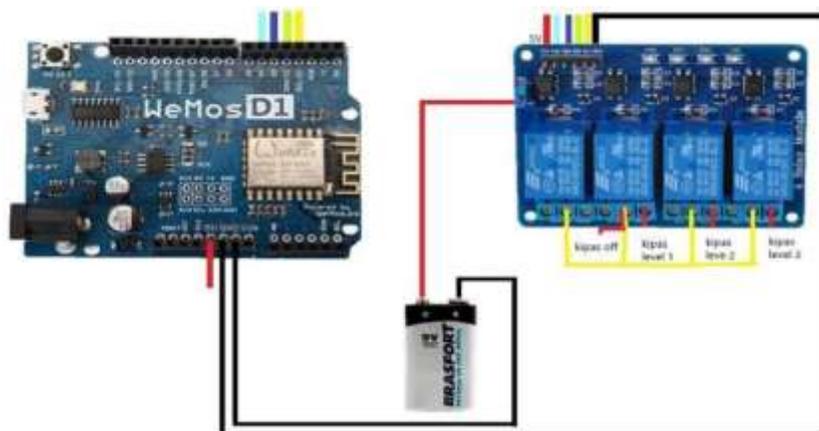
## METODE PENELITIAN

### Arsitektur Sistem

Tampilan arsitektur sistem pada penelitian ini tampak pada Gambar 3, input dari webcam laptop yang berfungsi untuk menampilkan video pergerakan tangan si pengguna, keluaran dari *webcam* selanjutnya diproses secara visi komputer oleh *library* MediaPipe untuk deteksi pergerakan serta bentuk dari gestur jari kedua tangan. Nilai dari hasil komputasi deteksi bentuk gestur jari tangan dari MediaPipe berikutnya dikirim ke MQTT *Broker*, yakni HiveMQ. Nilai data dari HiveMQ sebagai *publisher* di-*publish* ke modul WeMos-D1R2 yang berfungsi sebagai *subscriber* dengan topik yang sama, selanjutnya dilakukan aksi pada *relay* untuk menghidupkan, mematikan kipas angin serta mengontrol *level* kecepatan putaran kipas angin berdasarkan gestur jari tangan dari jarak jauh. Desain perangkat keras dengan tampilan *wiring cable* dari *board* WeMos yang terhubung ke *relay* tampak pada Gambar 4, *relay* selanjutnya dihubungkan ke *level* kecepatan putaran kipas angin dengan kecepatan putaran 0 (kipas mati), *level* 1 (putaran kipas kecepatan 1), *level* 2 (putaran kipas kecepatan 2), dan *level* 3 (putaran kipas kecepatan 3).



Gambar 3. Arsitektur sistem



Gambar 4. Perancangan sistem perangkat keras

### Setting MQTT Broker

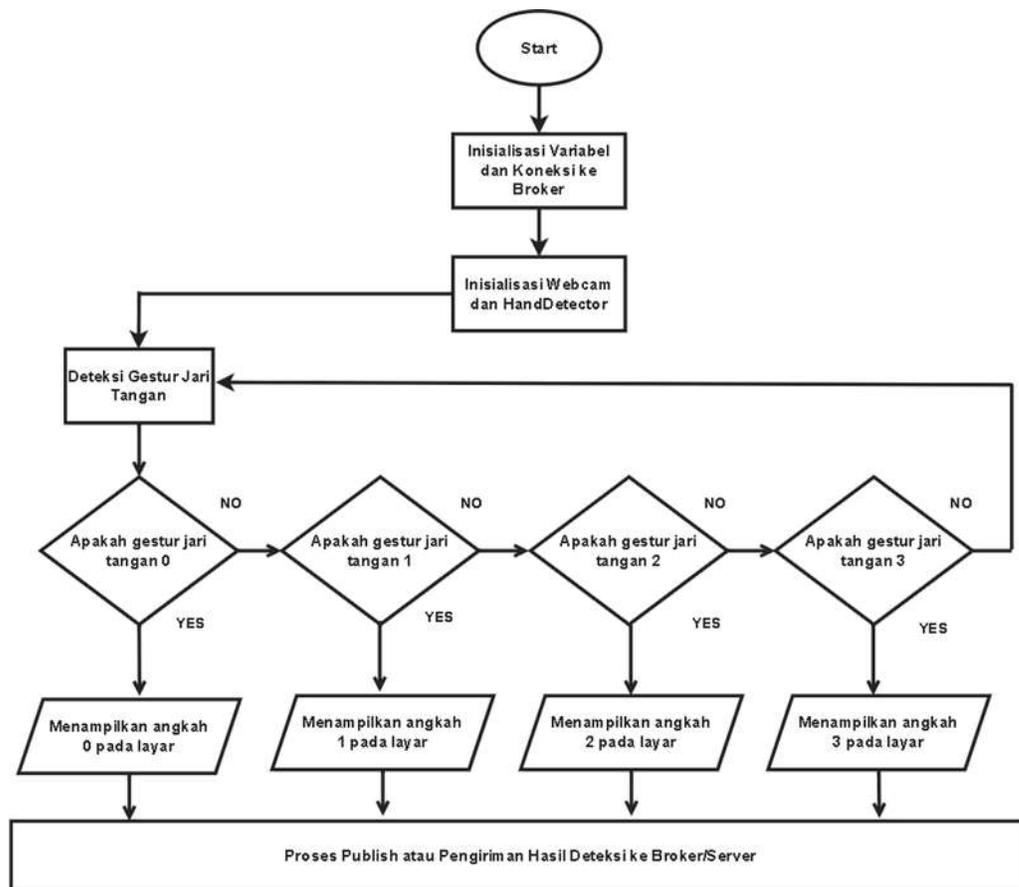
MQTT *Broker* seperti tampak pada Gambar 5, berfungsi sebagai server yang bertanggung jawab untuk proses penerimaan dan meneruskan data antara *subscriber* dan *publisher* dengan atribut yang harus sama seperti: *topic*, *username*, *password*, *port*, *mqttservice* dan beberapa atribut lainnya. MQTT *Websocket Client* pada Broker HiveMQ merujuk pada klien yang menggunakan *Websocket* untuk berkomunikasi dengan *broker* HiveMQ menggunakan protokol MQTT. Berdasarkan pada Gambar 5 atribut yang disetting pada broker HiveMQ antara lain adalah *Host* (*mqtt-dashboard.com*), *Port* (8884) dan *Topic*-nya "kipas\_angin". Secara umum *Host*, *Port*, dan beberapa atribut lainnya seperti *ClientID*, *SSL* secara otomatis ditetapkan oleh MQTT *Websocket Client* itu sendiri sebagai *default*.



Gambar 5. Alur sistem MQTT *Broker*

### Program Perangkat Lunak

Berdasarkan pada Gambar 6, program melakukan inisialisasi dan koneksi ke *broker* yang berfungsi sebagai *server* terlebih dahulu. *Webcam* laptop melakukan deteksi bentuk gestur jari tangan apakah bentuk gestur jari tangan berjumlah nol, satu, dua, atau tiga jika tidak ada jari tangan yang terdeteksi maka program melakukan proses *looping* kembali ke awal proses deteksi gestur jari tangan. Setelah proses deteksi mendapatkan nilai dilanjutkan dengan menampilkan nilai tersebut ke layar laptop sebagai indikator bahwa gestur jari tangan berhasil terdeteksi. Hasil nilai yang tampil pada layar laptop kemudian di *publish* ke MQTT *Broker* untuk selanjutnya diproses oleh client yang berfungsi sebagai *subscriber*.

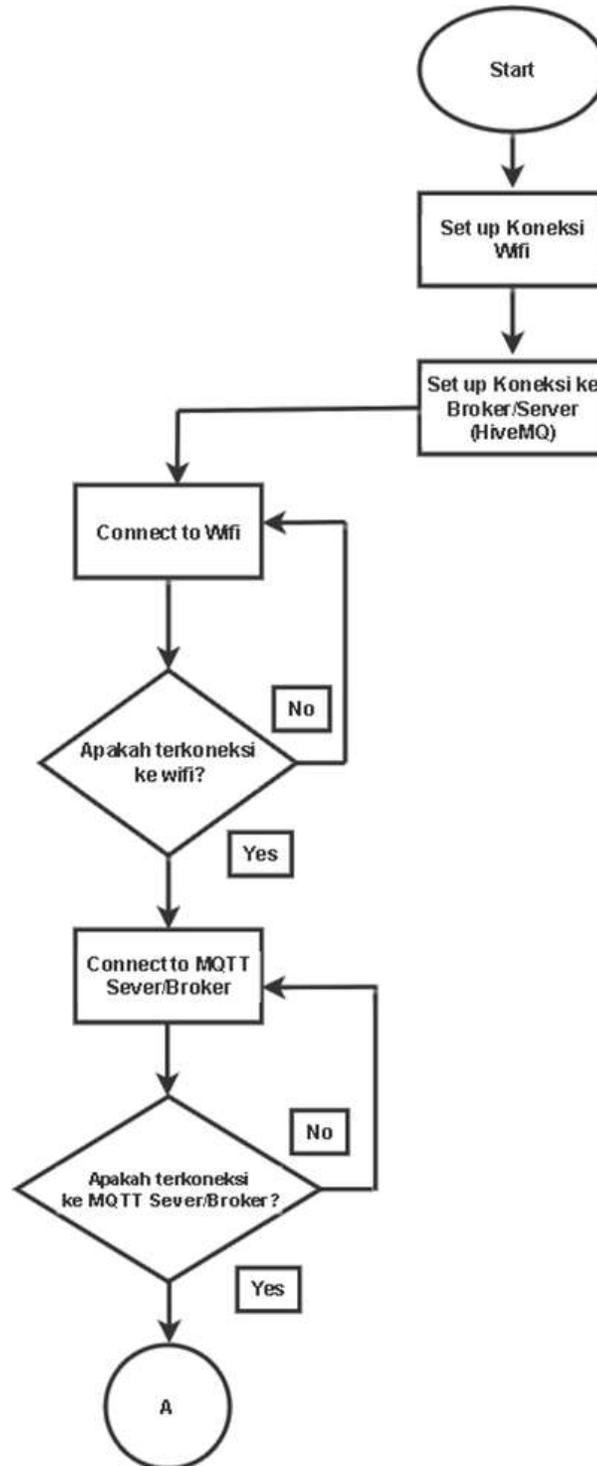


**Gambar 6. Flowchart proses deteksi gestur jari tangan**

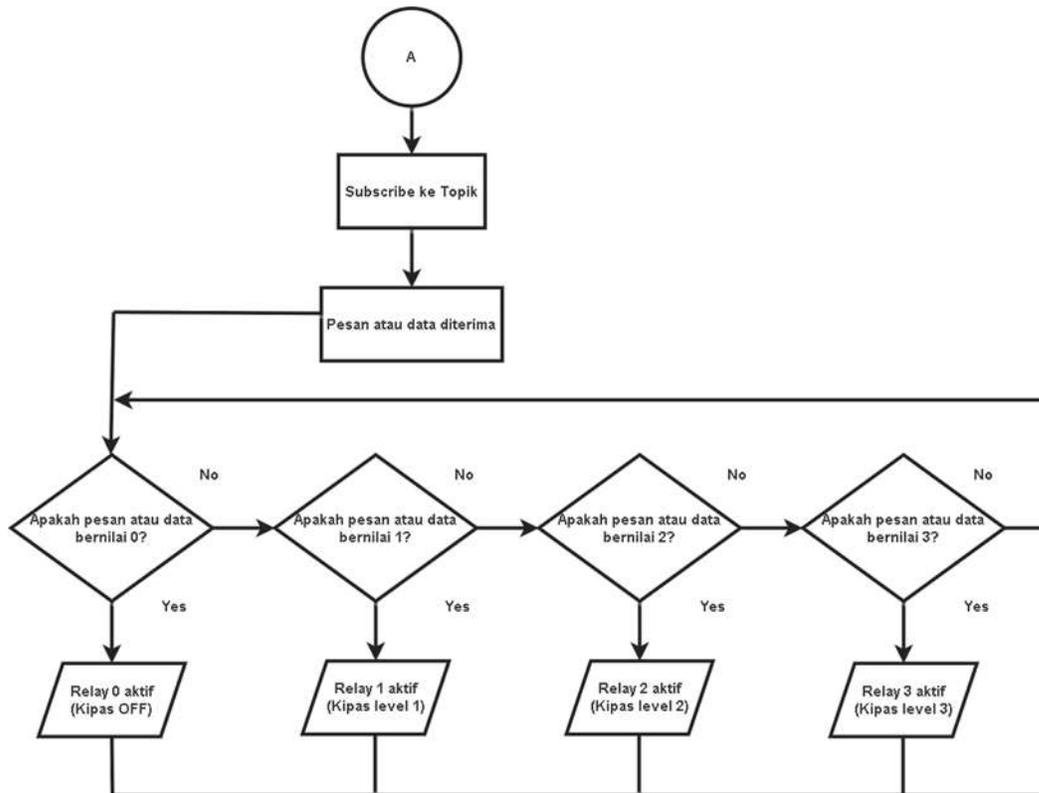
**Flowchart Konfigurasi MQTT Broker**

Flowchart program sebagaimana tampak pada Gambar 7 di bawah, memungkinkan pengendalian perangkat melalui pesan MQTT yang dikirimkan dengan topik "kipas\_angin". Pesan tersebut diinterpretasikan oleh WeMos-D1R2 untuk melakukan aksi pada relay sesuai dengan pesan yang diterima. Pada Gambar 7, dapat dilihat alur program pada WeMos-D1R2 pada saat pertama kali dijalankan. Program dimulai dengan membangun koneksi ke Wifi yang tersedia, jika koneksi ke Wifi berhasil maka proses berikutnya dilanjutkan dengan koneksi ke broker atau server, yakni HiveMQ, jika koneksi ke Wifi dan broker gagal maka program melakukan proses koneksi sampai koneksi tersebut berhasil terhubung.

Setelah koneksi Wifi dan broker berhasil, proses dilanjutkan dengan melakukan subscribe dengan topik "kipas\_angin" untuk proses komunikasi antar publisher, broker dan subscriber. Dengan melakukan subscribe atau berlangganan ke topik "kipas\_angin", proses dilanjutkan dengan menerima pesan atau data dari broker. Program terus berjalan dalam loop utama untuk memeriksa koneksi MQTT dan memproses pesan yang diterima seperti terlihat pada Gambar 8.



Gambar 7. Flowchart konfigurasi broker

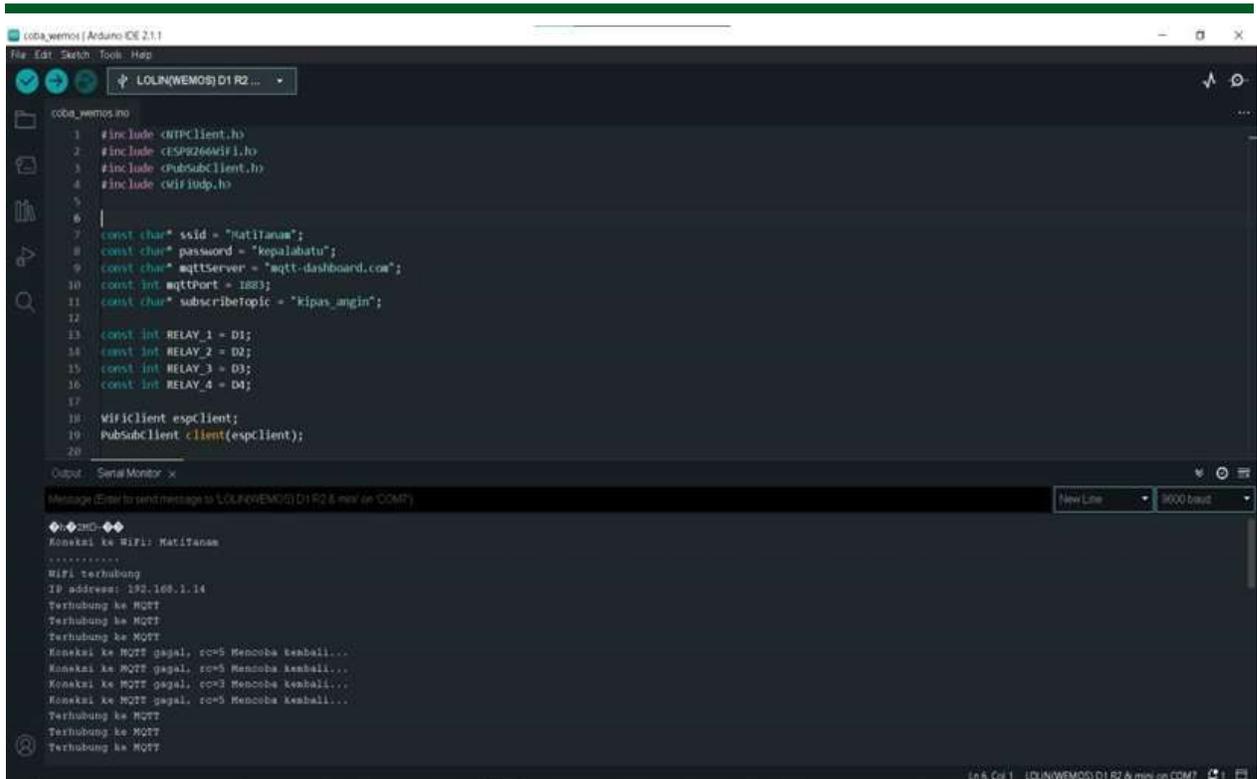


Gambar 8. Flowchart program WeMos-D1R2

## HASIL DAN PEMBAHASAN

### Hasil Pengujian Pengiriman dan Penerimaan Pesan antara Perangkat dengan MQTT Broker

Tampilan modul WeMos-D1R2 berhasil terhubung ke Wi-Fi dan juga terkoneksi ke MQTT Broker yang dituju tampak pada Gambar 9, maka proses penerimaan dan pengontrolan data dari *broker* dan *client* sebagai *subscriber* siap dieksekusi. Data yang diproses secara visi komputer dengan menerapkan metode Mediapipe kemudian diteruskan ke broker yang selanjutnya dikirimkan ke Modul WeMos-D1R2 sebagai *client* yang mengendalikan kipas angin. *Output* dari proses visi komputer yang dikirimkan ke *broker* adalah data gestur jari tangan yang terdeteksi terangkat keatas, seperti yang terlihat pada Gambar 10. Hasil pengujian yang telah dilakukan dapat dikatakan, data yang diterima oleh Modul WeMos-D1R2 sudah sesuai dengan data atau hasil yang diproses secara visi komputer dan dikirimkan ke MQTT *Broker*. Ketepatan waktu data terkirim ke broker dan data diterima oleh Modul WeMos-D1R2 dapat dibilang *real-time* sesuai harapan peneliti. waktu penerimaan pesan atau data oleh WeMos-D1R2 seperti yang dapat dilihat pada Tabel 1, dapat dikatakan cukup *real-time*. Hal ini dibuktikan dengan cara membandingkan waktu data yang masuk di *broker* dengan waktu penerimaan data pada sisi WeMos-D1R2 yang sama.



Gambar 9. Koneksi Wifi dan Broker



Gambar 10. Proses hasil deteksi dengan visi komputer

### Hasil Uji Pengontrolan *Level* Kecepatan Putaran Kipas Angin Secara Visi Komputer Dari Jarak Jauh Menggunakan Protokol MQTT

Percobaan selanjutnya adalah pengambilan data dari 5 orang pengguna dengan perubahan jarak antara pengguna dengan webcam sepanjang 100 cm, 150 cm dan 200 cm, sebagaimana tampak pada Tabel 2, hasilnya mendapatkan bentuk gestur jari yang tepat adalah 60 dengan total kesalahan deteksi adalah 0. Hal ini mengindikasikan bahwa proses *computer vision* yang dilakukan berjalan sesuai dengan harapan penulis. Proses mengontrol *level* kecepatan putaran kipas angin tidak semuanya berjalan seperti yang ditetapkan. Dapat dilihat pada percobaan pengontrolan *level* kecepatan putaran kipas angin yang dilakukan oleh setiap subjek, *output* gestur jari 0, 1, 2, 3 yang mewakili *level* kecepatan 0 (*off*), 1 (*high*), 2 (*medium*), dan 3 (*low*) memiliki jeda atau *delay* yang panjang sebelum kipas berada pada keadaan *On* atau *Off*.

**Tabel 1. Pengujian seberapa *real-time* kontrol *level* kecepatan kipas angin**

No	Broker HiveMQ	Waktu data masuk	WeMos-D1R2	Waktu data masuk	Status kipas		Keterangan
					On	Off	
1	1	11:19:54	1	19:19:54	√	-	Kipas level 1
2	2	11:20:01	-	-	-	-	Koneksi gagal
3	2	11:20:02	2	11:20:02	√	-	Kipas level 2
4	1	11:20:08	1	11:20:08	√	-	Kipas level 1
5	3	11:20:14	-	-	-	-	Koneksi gagal
6	3	11:20:40	3	11:20:40	√	-	Kipas level 3
7	2	11:20:42	2	11:20:42	√	-	Kipas level 2
8	3	11:20:54	3	11:20:54	√	-	Kipas level 3
9	0	11:20:55	0	11:20:55	-	√	Kipas level 0
10	1	11:21:04	1	11:21:04	√	-	Kipas level 1
11	2	11:21:15	-	-	-	-	Koneksi gagal
12	2	11:21:20	-	-	-	-	Koneksi gagal
13	2	11:21:25	-	-	-	-	Koneksi gagal
14	2	11:21:46	-	-	-	-	Koneksi gagal
15	0	11:22:06	0	11:22:06	-	√	Kipas level 0
16	2	11:22:11	2	11:22:11	√	-	Kipas level 2
17	3	11:22:17	3	11:22:17	√	-	Kipas level 3
18	2	11:22:18	2	11:22:18	√	-	Kipas level 2
19	1	11:22:20	-	-	-	-	Koneksi gagal
20	1	11:22:32	1	11:22:32	√	-	Kipas level 1
21	3	11:22:32	3	11:22:32	√	-	Kipas lev
22	3	11:22:36	3	11:22:36	√	-	Kipas leve
23	3	11:22:37	3	11:22:37	√	-	Kipas level

No	Broker HiveMQ	Waktu data masuk	WeMos-D1R2	Waktu data masuk	Status kipas		Keterangan
					On	Off	
24	0	11:22:40	-	-	-	-	Koneksi gagal
25	0	11:22:47	0	11:22:47	-	√	Kipas level 0
26	0	11:22:48	0	11:22:48	-	√	Kipas level 0
27	1	11:22:57	1	11:22:57	√	-	Kipas level 1
28	2	11:23:02	2	11:23:02	√	-	Kipas level
29	3	11:23:17	2	11:23:17	√	-	Kipas level 3
30	0	11:23:20	-	-	-	-	Koneksi gagal

**Tabel 2. Pengujian putaran kipas angin melalui *computer vision***

No	Nama	Jarak (cm)	Gestur	level Kecepatan Kipas				Hasil deteksi		FPS	Keterangan
				0	1	2	3	Benar	Salah		
1	Fredri	100	0	√				√	-	22	Berhasil
			1		√			√	-	24	Berhasil
			2			√		√	-	21	Berhasil
			3				√	√	-	23	Berhasil
		150	0	√				√	-	31	Berhasil
			1		√			√	-	31	Berhasil
			2			√		√	-	31	Berhasil
			3				√	√	-	31	Kipas delay
		200	0	√				√	-	31	Kipas delay
			1		√			√	-	31	Kipas delay
			2			√		√	-	31	Berhasil
			3				√	√	-	31	Berhasil
2	Yosea	100	0	√				√	-	31	Kipas delay
			1		√			√	-	31	Kipas delay
			2			√		√	-	31	Kipas delay
			3				√	√	-	31	Kipas delay
		150	0	√				√	-	31	Kipas delay
			1		√			√	-	32	Berhasil
			2			√		√	-	31	Kipas delay
			3				√	√	-	31	Kipas delay
		200	0	√				√	-	31	Kipas delay
			1		√			√	-	31	Kipas delay
			2			√		√	-	31	Kipas delay
			3				√	√	-	31	Berhasil

### Hasil Percobaan Tingkat Akurasi Proses Pengontrolan Kipas Angin Melalui Deteksi Bentuk Gestur Jari Tangan Secara Jarak Jauh

Telah dilakukan percobaan untuk mencari seberapa besar tingkat akurasi dari proses pengontrolan kecepatan putaran kipas angin melalui deteksi bentuk gestur jari tangan yang dilakukan menggunakan *library* Mediapipe sebagaimana terlihat pada Tabel 3. Dari pengujian yang dilakukan untuk mendapatkan tingkat akurasi yang tinggi pada Tabel 3, didapatkan gestur jari 0 memiliki tingkat akurasi sebesar 100% dengan FPS 28.6, sedangkan untuk gestur jari 2 dan gestur jari 3 memiliki tingkat akurasi 99% dengan masing-masing FPS sebesar 28.6 dan 31.4, dan terakhir diikuti oleh gestur jari 2 dengan tingkat akurasi 94% dengan FPS sebesar 29.9.

**Tabel 3. Pengujian akurasi gestur jari tangan**

No	Nama	Jarak (cm)	Gestur 10 kali percobaan	Hasil deteksi		Akurasi (%)	FPS	Keterangan
				Benar	Salah			
1	Fredri	100	0	10	0	100	31	Akurat
			1	10	0	100	31	
			2	10	0	100	31	
			3	10	0	100	29	
			0	10	0	100	28	
			1	10	0	100	27	
			2	10	0	100	26	
			3	10	0	100	26	
2	Yosea	100	0	10	0	100	21	Kurang Akurat
			1	8	2	80	22	
			2	10	0	100	21	
			3	10	0	100	21	
			0	10	0	100	21	
			1	8	2	80	22	
			2	10	0	100	22	
			3	10	0	100	22	
3	Niol	100	0	10	0	100	22	Kurang Akurat
			1	8	2	80	22	
			2	10	0	100	22	
			3	10	0	100	32	
			0	10	0	100	33	
			1	10	0	100	33	
			2	9	1	90	32	
			3	10	0	100	32	
4	Dani	100	0	10	0	100	33	Hampir Akurat
			1	10	0	100	33	
			2	10	0	100	33	
			3	10	0	100	33	
			0	10	0	100	32	
			1	10	0	100	33	
			2	9	1	90	33	
			3	9	1	90	33	

No	Nama	Jarak (cm)	Gestur 10 kali percobaan	Hasil deteksi		Akurasi (%)	FPS	Keterangan
				Benar	Salah			
5	Abet	100	0	10	0	100	33	Akurat
			1	10	0	100	33	
			2	10	0	100	33	
			3	10	0	100	33	
			0	10	0	100	33	
			1	10	0	100	33	
			2	10	0	100	33	
			3	10	0	100	33	
Rata-rata gestur 0						100	28.6	
Rata-rata gestur 1						94	29.9	
Rata-rata gestur 2						99	28.6	
Rata-rata gestur 3						99	31.4	

## KESIMPULAN

Kesimpulan hasil percobaan yang dilakukan secara empiris pada penelitian ini, sebagai berikut: pengontrolan kipas angin secara jarak jauh melalui deteksi bentuk gestur jari tangan secara *computer vision* berbasis IoT dengan protokol MQTT dapat dilakukan secara *realtime*, pada jarak 100 cm: gestur 0 akurasi 100% - FPS 28.6, gestur 1 akurasi 94% - FPS 29.9, gestur 2 akurasi 99% - FPS 28.6, gestur 3 akurasi 99% - FPS 31.4. Adapun pada jarak 150 cm deteksi bentuk gestur jari tangan 0, 1, 2, dan 3 dari lima orang akurasi 100%, jarak 200 cm, deteksi bentuk gestur jari tangan 0 akurasi 80%, gestur jari tangan 1 akurasi 80%, sedangkan gestur jari tangan 2 dan 3 akurasi 100%. Hasil pengujian pada jarak lebih dari 200 cm pada lima orang, deteksi bentuk gestur jari tangan 0 akurasi 80%, gestur jari tangan 1 akurasi 80%, gestur jari tangan 2 akurasi 100%, dan gestur jari tangan 3 akurasi 90%. Pengembangan penelitian lebih lanjut: pengujian menggunakan *library* Mediapipe diperlukan spesifikasi perangkat keras yang cukup tinggi, mikrokontroler yang mendukung komunikasi secara *wireless* dan *real-time* dan MQTT Broker memerlukan fitur pengamanan yang kuat agar data yang dikirim tidak hilang.

## DAFTAR PUSTAKA

- [1] Andy (2019) *WeMos D1 R2 Wifi Arduino Development Board*. Available at: <https://andydharma.com/wemos-d1-r2-wifi-arduino-development-board/> (Accessed: 20 July 2023).
- [2] Fakhrudin, M. A. (2023) *TA: Sistem Deteksi Gestur Jari Tangan menggunakan Mediapipe dan Faster-RCNN untuk Mengontrol Kecepatan Kipas Angin*. Universitas Dinamika. Available at: <https://repository.dinamika.ac.id/id/eprint/6862/>.
- [3] Fifit Alfiah, Budi Rahman and Imelda (2020) 'Control System prototype Smart Home IoT based with MQTT method using Google Assistant', *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 4(2), pp. 303–310. doi: 10.29207/resti.v4i2.1721.
- [4] Hernando, K. (2023) *TA: Implementasi Deep Learning untuk Visi Komputer sebagai Mouse Virtual menggunakan Mediapipe dan Faster RCNN*. Universitas Dinamika. Available at: <https://repository.dinamika.ac.id/id/eprint/6887/>.
- [5] Kukil (2022) *No Title, Introduction to MediaPipe*. Available at:

- <https://learnopencv.com/introduction-to-mediapipe/> (Accessed: 25 July 2022).
- [6] Lugaresi, C. *et al.* (2019) 'MediaPipe: A Framework for Building Perception Pipelines'. Available at: <http://arxiv.org/abs/1906.08172>.
- [7] Nautica, M. R. P. (2022) *TA : Hand Gesture Detection sebagai Alat Bantu Ajar Berhitung menggunakan Mediapipe dan Convolutional Neural Network secara Realtime*. Universitas Dinamika. Available at: <https://repository.dinamika.ac.id/id/eprint/6650/>.
- [8] R, H. (2022) 'Optimized Hand Gesture Based Home Automation for Feebles', *International Journal for Research in Applied Science and Engineering Technology*, 10, pp. 3433–3441. doi: 10.22214/ijraset.2022.45737.
- [9] Sobron, M. and Lubis (2021) 'Implementasi Artificial Intelligence Pada System Manufaktur Terpadu', *Seminar Nasional Teknik (SEMNASTEK) UISU*, 4(1), pp. 1–7. Available at: <https://jurnal.uisu.ac.id/index.php/semnastek/article/view/4134>.
- [10] Suprianto, D. *et al.* (2019) 'Microcontroller Arduino Untuk Pemula (Disertai Contoh-contoh Projek Menarik)', *Jurnal Jasakom*, 1(1), pp. 256–261. Available at: <https://www.researchgate.net/publication/335219524>.
- [11] Susanto, F., Prasiani, N. K. and Darmawan, P. (2022) 'IMPLEMENTASI INTERNET OF THINGS DALAM KEHIDUPAN SEHARI-HARI', *Jurnal Imagine*, 2(1), pp. 35–40. doi: 10.35886/imagine.v2i1.329.
- [12] Szeliski, R. (2022) *Computer Vision: Algorithms and Applications*. 2nd Editio. Springer. Available at: <http://szeliski.org/Book/>.