KONTROL *LEVEL* KECEPATAN PUTARAN KIPAS ANGIN MELALUI DETEKSI BENTUK GESTUR JARI TANGAN BERBASIS IoT

LAPORAN TUGAS AKHIR

Program Studi
Teknik Komputer

Oleh:
Fredi Wakerkwa
19410200033

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2023

KONTROL *LEVEL* KECEPATAN PUTARAN KIPAS ANGIN MELALUI DETEKSI BENTUK GESTUR JARI TANGAN BERBASIS IoT

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Teknik

Disusun Oleh:

Nama : Fredi Wakerkwa

NIM 19410200033

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2023

KONTROL *LEVEL* KECEPATAN PUTARAN KIPAS ANGIN MELALUI DETEKSI BENTUK GESTUR JARI TANGAN BERBASIS IoT

Dipersiapkan dan disusun oleh:

Fredi Wakerkwa

NIM: 19410200033

Telah diperiksa, dibahas, dan disetujui oleh dewan Pembahas

Pada: 17 Juli 2023

Susunan Dewan Pembahas

Pembimbing:

I. **Heri Praktikno, M.T., MTCNA., MTCRE.**
NIDN 0716117302

Digitally signed by Heri Pratikno, M.T.
Date: 2023.08.03 12:42:31 +07'00'

II. **Weny Indah Kusumawati, S.Kom., M.MT.**
NIDN 0721047201

Universitas Dinamika
2023.08.03 12:25:28 +07'00'

Penguji:

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

Digitally signed by Musayyanah
DN: cn=Musayyanah, o=Universitas Dinamika, ou=S1 Teknik Komputer, email=musayyanah@dinamika.ac.id, c=ID
Date: 2023.08.03 13:44:26 +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: 2023.08.04 11:02:10 +07'00'

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

NIDN: 0731017601

Dekan Fakultas Teknologi dan Infromatika

UNIVERSITAS DINAMIKA

"Terangnya pelita dimalam yang gelap gulita bagaikan semangat bapak dan mama yang tidak pernah sirna dan terus bercahaya"

-Fredi Wakerkwa-

Terima kasih saya ucapkan kepada semua orang tanpa terkecuali, yang terus memberikan saya kesempatan untuk berada ditengah-tengah kalian dan memberi saya kesempatan untuk belajar hal-hal baru, yang belum pernah saya ketahui sebelumnya. Pencapaian ini tidak akan pernah berhasil jika tidak ada campur tangan dari Tuhan, keluarga, dosen pembimbing, serta teman-teman sekalian.

PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, saya:

Nama	: **Fredi Wakerkwa**
NIM	: **19410200033**
Program Studi	: **S1 Teknik Komputer**
Fakultas	: **Fakultas Teknologi dan Informatika**
Jenis Karya	: **Laporan Tugas Akhir**
Judul Karya	: **KONTROL *LEVEL* KECEPATAN PUTARAN KIPAS ANGIN MELALUI DETEKSI BENTUK GESTUR JARI TANGAN BERBASIS IoT**

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

Fredi Wakerkwa
NIM: 19410200033

ABSTRAK

Teknologi berkembang sangat cepat seiring berjalannya waktu, *Artificial Intelligence* salah satunya. Hampir disetiap hari dan tempat menggunakan teknologi ini, yang didalamnya terdapat *Artificial Intelligence*. Teknologi *computer vision* merupakan cabang ilmu dari kecerdasan buatan yang mampu memproses data visual. Selain itu, perkembangan mikrokontroler juga tidak kalah penting guna mempermudah pekerjaan manusia. Pada penelitian Tugas Akhir ini dilakukan pengontrolan kipas angin mengunakan metode atau *library* Mediapipe yang dijalankan diatas *computer vision* untuk mendeteksi gestur jari tangan yang terdeteksi dan mengirimkan data hasil deteksi melalui MQTT *Broker* untuk dikontrol menggunakan Modul WeMos-D1R2. Studi ini membuktikan bahwa kipas angin dapat dikontrol secara jarak jauh melalui deteksi gestur jari tangan menggunakan *computer vision* dan IoT dengan protokol MQTT secara real-time. Hasil percobaan menunjukkan akurasi deteksi gestur jari tangan pada jarak 100 cm, di mana gestur 0 memiliki akurasi 100% dengan FPS 28.6, gestur 1 memiliki akurasi 94% dengan FPS 29.9, gestur 2 memiliki akurasi 99% dengan FPS 28.6, dan gestur 3 memiliki akurasi 99% dengan FPS 31.4. Selain itu, hasil pengujian perubahan pada jarak 150 cm menunjukkan akurasi 100% untuk deteksi gestur 0, 1, 2, dan 3 pada lima orang. Pada jarak 200 cm, akurasi deteksi gestur 0 dan 1 adalah 80%, sedangkan gestur 2 dan 3 tetap memiliki akurasi 100%. Pada jarak lebih dari 200 cm, gestur 0 dan 1 memiliki akurasi 80%, gestur 2 memiliki akurasi 100%, dan gestur 3 memiliki akurasi 90%.

Kata Kunci: *Computer Vision*, Mediapipe, MQTT *Broker*, *Publisher*, *Subscriber*. WeMos-D1R2.

KATA PENGANTAR

Dengan penuh rasa syukur, penulis ingin mengucapkan terima kasih kepada Tuhan Yang Maha Esa atas berkat-Nya yang melimpah, yang memungkinkan penulis menyelesaikan laporan tugas akhir dengan judul "Kontrol *Level* Kecepatan Putaran Kipas Angin melalui Deteksi Bentuk Gestur Jari Tangan Berbasis IoT". Penulis merasa sangat berterima kasih atas banyaknya bantuan yang diterima selama proses pengerjaan Laporan Tugas Akhir ini. Dalam kesempatan ini, penulis ingin mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan, antara lain:

1. Allah, 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 materiil, 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 banyak terima kasih untuk setiap dorongan positif dan saran yang diberikan sehinggap penulis dapat menyelesaikan Tugas Akhir.
6. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku dosen pembimbing Penulis mengucapkan terima kasih atas bimbingan, kesempatan, dan panduan baik lisan maupun tertulis yang diberikan, yang telah membantu penulis dalam menyelesaikan Tugas Akhir ini.
7. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku dosen pembimbing Penulis sangat berterima kasih atas bantuan berupa perhatian dan bimbingan terhadap tantangan yang dihadapi penulis dalam menyelesaikan Tugas Akhir ini.

8. Seluruh rekan–rekan S1 Teknik Komputer angkatan 2019 yang telah memberikan dukungan dan semangatnya untuk membantu penulis menyelesaikan laporan Tugas Akhir ini.
9. Seluruh pihak yang tidak dapat disebutkan satu per satu yang telah memberikan dukungan serta bantuan dalam segala bentuk yang akhirnya terselesaikannya laporan Tugas Akhir ini.

Surabaya, 17 Juli 2023

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

Halaman

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pada beberapa tahun belakangan ini, teknologi mengalami kemajuan yang begitu signifikan, terutama dalam bidang IoT (*Internet of Things),* AI (*Artificial Intelligence*), dan *Computer Vision* (CV). IoT memungkinkan pengguna untuk menghubungkan dan mengontrol perangkat elektronik dari jarak jauh, sehingga memudahkan pengguna dalam mengelola dan mengawasi perangkat mereka dari mana saja. Sementara itu, kemajuan dalam AI (*Artificial Intelligence*) dan CV (*Computer Vision*) telah memberikan kemampuan mesin untuk memproses dan memahami informasi dari gambar dan video. Hal ini memungkinkan aplikasi yang lebih luas dari teknologi ini dalam berbagai sektor, seperti pertanian, manufaktur, transportasi, dan medis. Selain itu, penggunaan mikrokontroler juga semakin banyak diterapkan dalam pengembangan sistem kontrol otomatis, yang dapat mempermudah kontrol dan pengawasan untuk sistem yang kompleks.

CV (*Computer Vision*) merupakan bidang studi tentang bagaimana komputer mampu mengambil, memproses, dan mempelajari informasi visual dari dunia nyata. Ini melibatkan penggunaan teknologi dan algoritma untuk mengubah citra atau video menjadi data yang dapat dipahami oleh komputer dan dimanfaatkan untuk berbagai aplikasi, seperti deteksi objek, pengenalan wajah, analisis citra medis, dan robotika (Szeliski, 2022).

Pengendalian sistem di era sekarang ini telah banyak menggunakan mikrokontroler untuk kebutuhan sehari–hari manusia seperti, *controlling*, *monitoring*, maupun *securing.* Tujuannya tidak lain untuk mempermudah pekerjaan dan meningkatkan kualitas hidup. 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) memberikan landasan yang kuat bagi peneliti untuk mengembangkan lebih lanjut konsep deteksi jari dalam pengontrolan kecepatan putaran kipas angin yang lebih efektif dan efisien dari jarak jauh. Sebagai pendukung penelitian, peneliti juga mengambil referensi jurnal yang berjudul

Optimized Hand Gesture Based Home Automation for Feebles (Hithaishi, 2022), tujuannya untuk mengembangkan sistem pengaturan perangkat rumah yang dapat dikendalikan melalui gerakan tangan dan pola penggunaan. Dari penelitian dan jurnal tersebut, keduanya dapat diimplementasikan dalam penelitian ini.

Pada Tugas Akhir kali ini, penulis akan melakukan kontrol nyala (*ON*), mati (*OFF*) serta kontrol 3 *level* kecepatan putaran kipas angin secara jarak jauh berbasis IoT menggunakan protokol MQTT. Adapun lingkungan IoT dibagun diatas papan pengembangan WeMos-D1R2 berbasis ESP8266, sedangkan untuk mengontrol *level* kecepatan kipas angin melalui proses deteksi bentuk gestur jari tangan kanan ataupun tangan kiri menggunakan kamera *webcam* laptop yang telah dideklarasikan didalam program. Proses deteksi pengenalan bentuk gestur jari tangan dilakukan oleh Mediapipe.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dirumuskan permasalahan pada Tugas Akhir ini sebagai berikut:

1. Bagaimana mengendalikan *level* kecepatan putaran kipas angin dari jarak jauh menggunakan *computer vision*?
2. Seberapa besar nilai *real-time* dari sistem pengontrolan *level* kecepatan putaran kipas angin secara jarak jauh berbasis *computer vision*?
3. Berapa besar tingkat akurasi proses kontroling *level* kecepatan kipas angin secara jarak jauh menggunakan protokol MQTT?
4. Pada jarak berapakah didapatkan jarak yang paling tinggi tingkat akurasinya antara *webcam* dan jari tangan?

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, cakupan pembahasan masalah dibatasi pada beberapa hal sebagai berikut:

1. Pendeteksian hanya dilakukan oleh salah satu gestur jari tangan saja.
2. Jumlah bentuk gestur jari setiap tangan hanya dapat mendeteksi semua jari mengenggam (0, kipas mati), salah satu jari berdiri (1, putaran kipas *level* 1),

dua jari berdiri (2, putaran kipas *level* 2) dan tiga jari berdiri (3, putaran kipas *level* 3).

3. Pencahayaan ruangan yang merata.

1.4 Tujuan

Berdasarkan latar belakang dan rumusan masalah yang telah dipaparkan, tugas akhir ini bertujuan untuk mencapai hal-hal berikut:

1. Mampu mengontrol *level* kecepatan putaran kipas angin dari jarak jauh menggunakan *computer vision*.
2. Dapat mengetahui besar tingkat akurasi proses kontroling *level* kecepatan kipas angin secara jarak jauh menggunakan protokol MQTT.
3. Dapat mengetahui jarak yang paling tinggi tingkat akurasinya dalam proses deteksi atau pengenalan bentuk gestur jari tangan antara *webcam* dan jari tangan.
4. Dapat mengetahui besar nilai *real–time* dari sistem pengontrolan *level* kecepatan putaran kipas angin secara jarak jauh berbasis *computer vision*.

1.5 Manfaat

Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut:

1. Bagi penulis, dapat memberikan dasar pemahaman terkait implementasi *computer vision* dengan penggabunggan teknologi berbasis IoT.
2. Bagi mahasiswa, menjadi pintu explorasi pengembangan bidang penerapan *computer vision* dalam ranah IoT.
3. Mempermudah pengontrolan perangkat elektronik secara jarak jauh tanpa menggunakan *sensor* tetapi hanya dengan menggunakan *webcam*.

BAB II
LANDASAN TEORI

2.1 Artificial Intelligence

Kecerdasan manusia dalam menyelesaikan masalah disebabkan oleh pengetahuan dan pengalaman yang dimilikinya. Pengetahuan ini didapatkan melalui proses belajar. Semakin banyak pengetahuan yang dimiliki, semakin baik manusia dalam menyelesaikan masalah. Namun, kecerdasan manusia tidak hanya terbatas pada pengetahuan saja. Manusia juga mempunyai kemampuan untuk berpikir secara rasional dan menggunakan pengetahuan serta pengalaman yang dimilikinya untuk mengambil keputusan yang tepat. Tanpa kemampuan berpikir yang baik, pengetahuan dan pengalaman tidak berguna dalam menyelesaikan masalah. Sebaliknya, tanpa pengetahuan dan pengalaman yang memadai, kemampuan berpikir manusia tidak mencukupi untuk menyelesaikan masalah dengan baik. Hal yang sama juga berlaku untuk mesin cerdas. Untuk bisa bertindak seperti manusia, mesin harus diberi pengetahuan yang cukup dan kemampuan berpikir yang baik (Lubis, 2021).

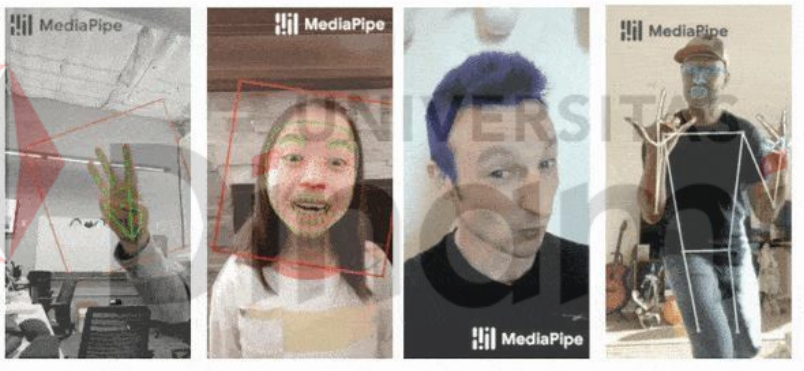
2.2 Python

Sejak awal munculnya pada tahun 1990-an, Python selalu menjadi opsi bahasa pemrograman yang paling populer di industri. Fakta tersebut juga ditunjukkan oleh survei dari RedMonk pada tahun 2021, yang menempatkan Python pada peringkat kedua sebagai bahasa pemrograman favorit para pengembang setelah sekitar 30 tahun sejak awal diluncurkan (algoritma, 2022). Python digunakan untuk mengembangkan aplikasi, membuat perintah komputer, dan melakukan analisis data. Sebagai bahasa pemrograman *general-purpose*, Python dapat digunakan untuk membuat program apa saja dan menyelesaikan berbagai masalah. Walaupun dianggap mudah dipelajari, Python termasuk bahasa pemrograman tingkat tinggi yang banyak dipilih oleh berbagai profesi, mulai dari *back-end developer*, IT, hingga data *scientist*.

Gambar 2.1 Bahasa pemrograman Python
(Sumber: (Python, 2023))

2.3 Mediapipe

MediaPipe merupakan sebuah *platform* yang digunakan untuk membuat pipa-pipa pembelajaran mesin yang dapat memproses data *time-series* seperti *video*, *audio*, dan lain-lain. *Platform* ini dapat dijalankan pada beberapa *platform* seperti *Desktop/Server*, Android, iOS, dan juga perangkat tersemat seperti Raspberry Pi dan Jetson Nano. *Platform* ini dirancang untuk memudahkan pengguna dalam membuat dan mengembangkan *pipeline* yang kompleks untuk memproses data yang bersifat dinamis dan memerlukan pemrosesan yang cepat (Kukil, 2022).



Gambar 2.2 Bentuk output Mediapipe
(Sumber: (Kukil, Introduction to MediaPipe, 2022))

2.4 Board WeMos-D1R2

WeMos-D1R1 merupakan sebuah papan pengembangan yang kompatibel dengan Arduino dan memiliki modul Wifi ESP8266 yang sudah terintegrasi. Papan pengembangan WeMos D1R2 ini merupakan papan dengan fitur WiFi yang paling terjangkau saat ini. Papan ini didasarkan pada chip ESP8266 yang mampu terhubung ke jaringan WiFi. Secara fisik, papan ini memiliki tampilan yang mirip dengan papan Arduino standar. Dimensi dan susunan pin pada papan ini sama persis dengan Arduino, sehingga papan ini dapat digunakan dengan semua perisai Arduino yang telah ada.

Gambar 2.3 *Board* WeMos-D1R2 berbasis ESP8266
(Sumber: (ANDY, WEMOS D1 R2 WIFI ARDUINO DEVELOPMENT BOARD, 2019))

Chip ESP8266 yang digunakan pada papan WeMos D1R2 memiliki fitur-fitur berikut:

1. CPU RISC 32-bit yang beroperasi pada kecepatan 80MHz
2. RAM instruksi sebesar 64Kb dan RAM data sebesar 96Kb
3. Memori *flash* sebesar 4MB!
4. Kemampuan Wi-Fi
5. Terdapat 16 pin GPIO
6. Mendukung komunikasi I2C dan SPI
7. Mendukung I2S
8. Terdapat 1 ADC

2.5 Arduino IDE

Arduino IDE adalah sebuah perangkat lunak khusus dirancang untuk membuat dan mengedit kode pemrograman atau *sketch* pada *board* Arduino. Dengan Arduino IDE, pengguna dapat dengan mudah menulis, mengunggah, dan memprogram *board* Arduino dengan bahasa pemrograman C/C++ (erintafifah, 2021). Sebagai alat pemrograman yang bersifat *open-source*, Arduino IDE juga dilengkapi dengan berbagai fitur dan fungsi yang memudahkan pengguna untuk melakukan berbagai tugas pemrograman. Arduino IDE juga dapat digunakan pada sistem operasi Windows, MacOS, dan Linux, sehingga dapat digunakan oleh pengguna dari berbagai *platform*.

Gambar 2.4 *Software* Arduino IDE
(Sumber: (Arduino IDE logo.svg, 2023))

2.6 IoT (Internet of Things)

Pengertian *Internet of Things* atau yang sering dikenal dengan istilah IoT menurut Fredy susanto (2022) adalah sistem *embedded* yang bertujuan untuk memperluas pemanfaatan dari konektivitas internet yang tersambung secara terus-menerus. Melalui konektivitas yang terus-menerus, perangkat IoT dapat mengumpulkan data tentang lingkungan sekitarnya, seperti suhu, kelembaban, lokasi, dan banyak lagi. Informasi ini dapat digunakan untuk mengoptimalkan penggunaan sumber daya, meningkatkan efisiensi, dan menghadirkan pengalaman yang lebih personal dan terhubung. Dengan kata lain, *Internet of Things* (IoT) memungkinkan perangkat dan objek-objek sehari-hari terhubung dan bekerja bersama untuk menciptakan kehidupan yang lebih efisien, nyaman, dan terhubung secara digital.

2.7 MQTT

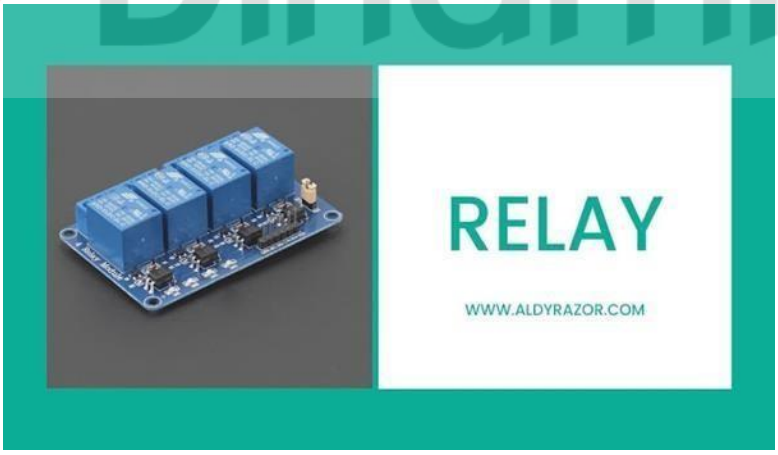
MQTT (*Message Queuing Telemetry Transport*) merupakan protokol yang beroperasi di atas TCP/IP. Prinsip kerjanya mirip dengan model *client-server*, tetapi dalam protokol ini lebih dikenal dengan istilah *publisher-subscriber* (Rahman & Imelda, 2020). MQTT adalah standar yang sering digunakan untuk komunikasi antar-perangkat *Internet of Things* (IoT) karena ukurannya yang kecil dan efisien. Protokol ini juga mendukung koneksi ke berbagai *platform* dan bahasa pemrograman. Dalam MQTT, *publisher* bertanggung jawab mengirimkan pesan ke *broker*, sedangkan *subscriber* berlangganan pesan dari *broker*.

Dalam MQTT, pesan dikirimkan dalam format yang sederhana dan ringkas, terdiri dari topik (*topic*) dan *payload*. Topik digunakan untuk mengidentifikasi jenis

pesan yang dikirim atau diterima, sedangkan *payload* berisi informasi yang sebenarnya. Kelebihan MQTT adalah kemampuannya untuk menjaga kehandalan pengiriman pesan, di mana pesan yang dikirimkan oleh *publisher* akan disimpan di *broker* dan disampaikan ke *subscriber* yang terhubung saat mereka tersedia, bahkan jika koneksi sementara terputus. Dengan mekanisme ini, MQTT dapat menangani kondisi jaringan yang tidak stabil atau perangkat yang berpindah-pindah, sehingga memastikan bahwa pesan-pesan penting dapat diantarkan dengan aman dan dapat diandalkan.

2.8 Modul Relay

Modul relay merupakan sebuah piranti yang menggunakan prinsip elektromagnetik untuk menggerakkan kontaktor guna memindahkan posisi *ON* ke *OFF* atau sebaliknya dengan memanfaatkan tenaga listrik. Fungsinya sebagai saklar listrik yang bekerja secara otomatis sesuai dengan perintah logika yang diberikan. Relay memiliki dua kondisi atau posisi umum, yakni *Normally Close* (NC) dan *Normally Open* (NO). Kondisi NC terjadi ketika relay dalam posisi tertutup karena tidak menerima arus listrik, sementara kondisi NO terjadi ketika relay dalam posisi terbuka karena menerima arus listrik (Razor, 2021).

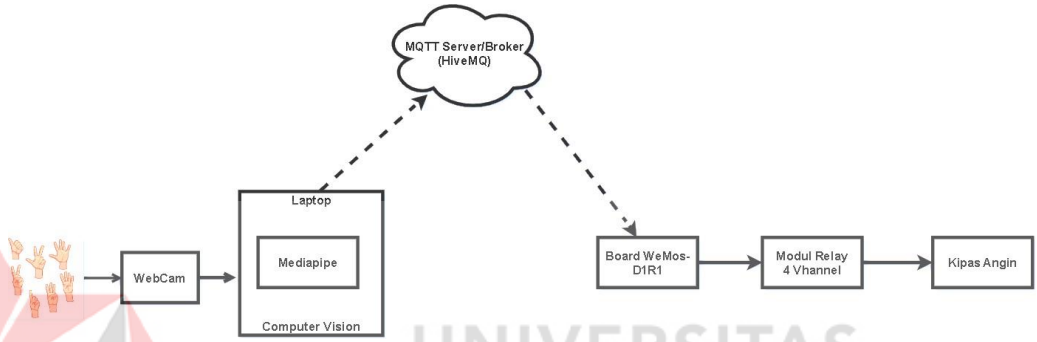


Gambar 2.5 Modul relay 4 *channel*
(Sumber: https: (Razor, Modul Relay Arduino, 2021))

BAB III
METODOLOGI PENELITIAN

Pada Bab 3 ini dijelaskan secara detail metode penelitihan yang dilakukan pada Tugas Akhir, mulai dari proses input hingga proses output serta penerapan protokol komunikasi yang diterapkan dalan Tugas Akhir ini.

3.1 Arsitektur Sistem



Gambar 3.1 Arsitektur sistem

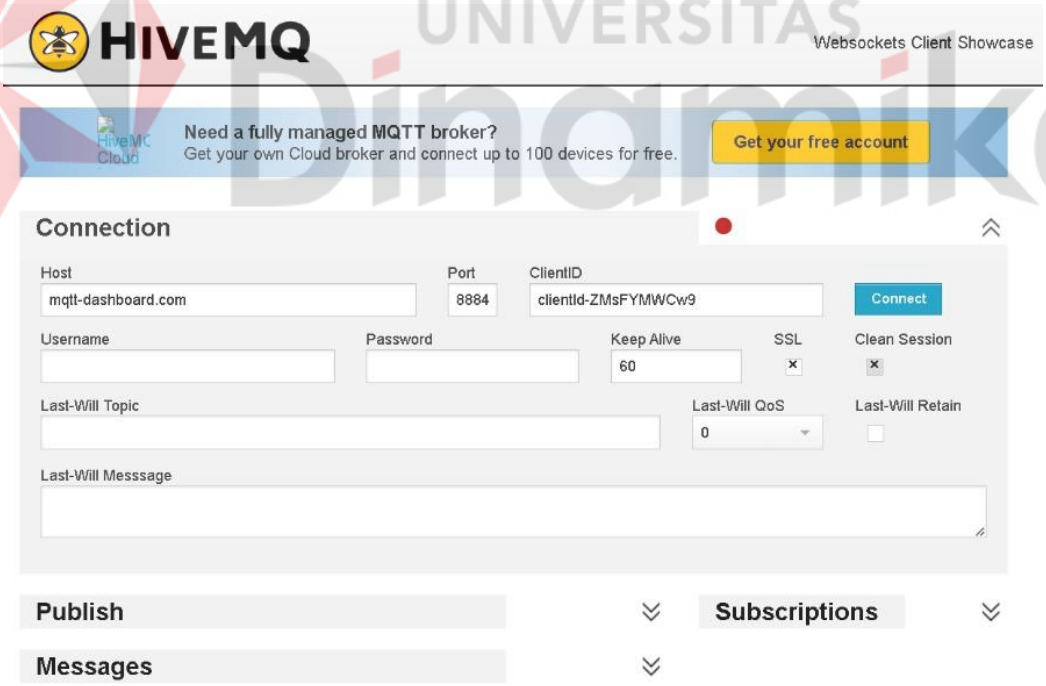
Seperti pada Gambar 3.1, inputan yang diinput adalah dari *webcam* laptop (*publisher*) yang sebelumnya dideklarasikan pada program deteksi gestur jari tangan. Inputan tersebut kemudia diproses oleh *computer vision* menggunakan *library* Mediapipe yang nilainya bisa dilihat di layar laptop. Kemudian, nilai tadi selain menampilkan keluaran pada layar laptop, juga sekalian mempublish keluaran dari *computer vision* ke MQTT *Broker*, yakin HiveMQ. Modul WeMos-D1R2 (*subscriber*) disisi lain, sebelum memberikan aksi pada relay untuk menghidupkan atau mematikan kipas angin, terlebih dahulu berlangganan atau *subscribe* ke topik yang sama, kemudian baru dapat melakukan pengontrolan *level* kecepatan putaran kipas angin menggunakan gestur jari tangan dari jarak jauh.

3.2 Setting MQTT Broker

Pada Tugas Akhir ini, MQTT *Broker* bertugas layaknya *server* yang bertanggung jawab untuk menerima dan meneruskan data.

Gambar 3.2 Alur sistem MQTT *Broker*

Agar proses pengiriman dari *publisher*, proses penerimaan dan pengiriman data oleh *broker*, serta proses penerimaan data oleh *subscriber* dapat berlangsung, terdapat syarat atau aturan yang harus dipenuhi terlebih dahulu. Syarat tersebut adalah data yang dikirimkan harus memiliki atribut yang sama dengan atribut yang dimiliki oleh *broker*, seperti *topic*, *username*, *password*, *port*, *mqttserver* serta beberapa atribut lainnya. Jika syarat-syarat tersebut telah terpenuhi, maka transaksi data antara *publisher*, *broker* dan *subscriber* dapat berlangsung tanpa masalah.

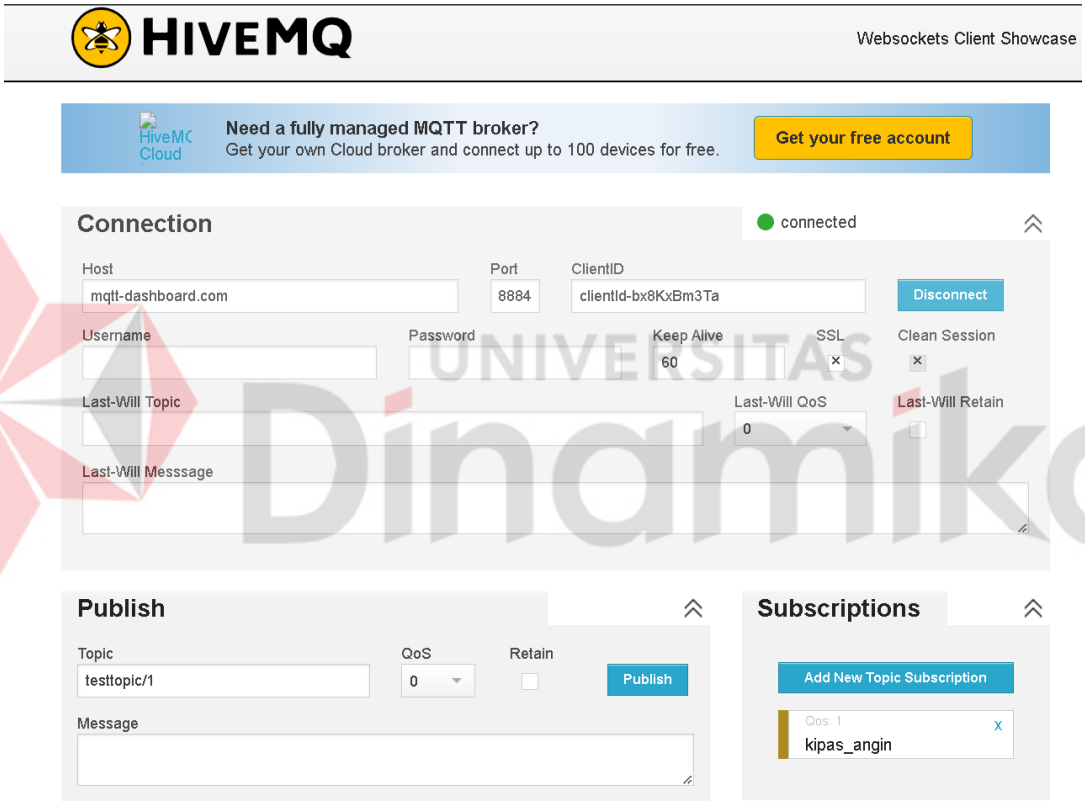


Gambar 3.3 Tampilan HiveMQ

Hal pertama yang dilakukan pertama kali sebelum mensetting *broker* adalah dengan mengunjungi URL atau alamat MQTT Websocket *Client*. MQTT

Websocket *Client* pada *Broker* HiveMQ itu sendiri merujuk pada klien yang menggunakan Websocket untuk berkomunikasi dengan *broker* HiveMQ menggunakan protokol MQTT.

Berdasarkan pada Gambar 3.3, atribut yang disetting pada *broker* HiveMQ antara lain adalah *Host* (mqtt-dashboard.com), *Port* (8884) dan *Topic* (kipas_angin). Pada umumnya, *Host*, *Port*, dan beberapa atribut lainnya seperti *ClientID*, SSL dan lain sebagainya, secara otomatis ditetapkan oleh MQTT Websocket *Client* itu sendiri sebagai *default* seperti terlihat pada Gambar 3.3. Namun, untuk *Topic* disesuaikan dengan kebutuhan.

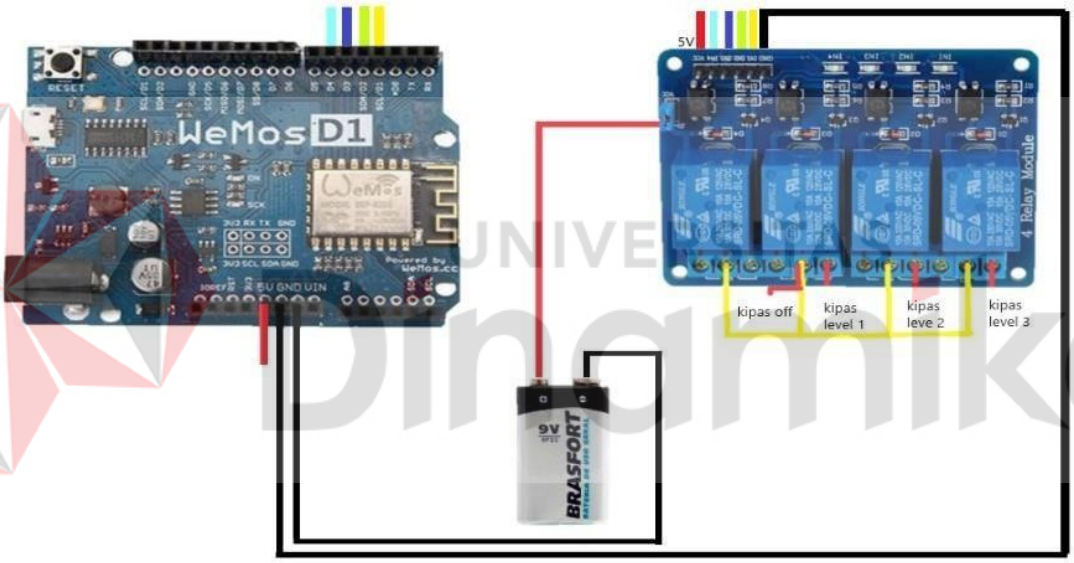


Gambar 3.4 Tampilan HiveMQ

Pada Gambar 3.4 merupakan tampilan MQTT Websocket *Client* setelah dikoneksikan. Pada bagian Subscriptions diisikan *topic* dengan nama "kipas_angin". Sedangkan, untuk QoS (*Quality of Service*) yang digunakan adalah QoS 1. Artinya, *Subscriber* menerima setiap pesan minimal satu kali. Jika pesan tidak berhasil diterima oleh *subscriber* pada saat itu, *broker* akan mengirimkan pesan kembali hingga diterima oleh *subscriber*.

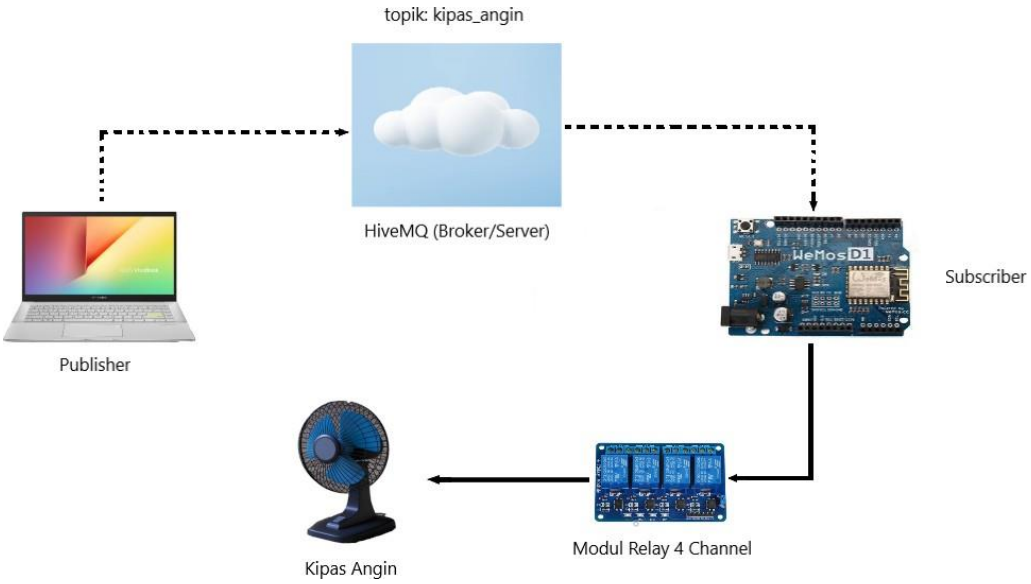
3.3 Desain Perangkat Keras

Gambar 3.5 merupakan proses *wiring cable* dari *board* WeMos- D1R2 untuk terhubung langsung ke relay. Relay kemudian hubungkan lagi ke *level* kecepatan putaran kipas angin yang dimulai dari *level* kecepatan putaran 0 (kipas mati), *level* 1 (putaran kipas *level* 1), *level* 2 (putaran kipas *level* 2), dan *level* 3 (putaran kipas *level* 3). NO yang berarti '*Normally Open'* dan NC yang berarti '*Normally Closed'* digunakan pada relay sebagai sakelar untuk memutus atau menghubungkan arus listrik. Saat relay tidak dialiri oleh listrik, pin COM dan NO akan terputus, sedangkan pin COM dan NC terhubung. Namun, saat relay hidup atau diberi tegangan, NO berubah menjadi NC dan NC akan berubah menjadi NO. Konfigurasi ini sering digunakan sebagai sakelar untuk mengontrol aliran listrik.



Gambar 3.5 Perancangan sistem *hardware*

Seperti dilihat pada Gambar 3.5, pin-pin pada *board* WeMos-D1R2 kemudian dihubungkan ke pin *trigger* pada relay 4 *channel*. Pertama, proses *wiring* dimulai dengan menghubungkan pin D1, D2, D3, dan D4 pada *board* WeMos ke pin inputan 1, 2, 3, dan 4 pada relay. Selain itu, pin 5V dan GND pada WeMos kemudian dihubungkan ke pin VCC dan GND pada relay sebagai sumber tegangan. Adapun penambahan tengangan eksternal menggunakan baterai 9V, yang mana untuk kabel positif baterai dihubungkan ke pin VCC tambahan relay dan kabel negatif dihubungkan ke GND pada *board* WeMos-D1R2.



Gambar 3.6 Alur keseluruhan sistem

Berdasarkan pada Gambar 3.6, dapat dijelaskan inputan atau masukan diperoleh dari *webcam* laptop yang telah dideklarasikan didalam program. Dalam proses *transferring* data, laptop terhubung ke MQTT *Broker* melalui koneksi jaringan atau *wireless* agar proses pengiriman data berhasil. Namun, dalam memberikan aksi ke kipas angin, relay terhubung langsung ke kipas angin dan *board* WeMos-D1R2 melalui kabel.

3.4 Metode Penerapan Format Data

Jenis data yang dikirimkan dari *computer vision* (*publisher*) ke MQTT *Broker* adalah data berupa karakter yang telah dikonversi. Seperti dapat dilihat pada program deteksi gestur jari tangan itu sendiri, dimana terdapat konversi jumlah jari yang terangkat ke dalam bentuk *string* sebelum ditampilkan di layar. Hal ini dilakukan dengan menghitung jumlah jari yang terangkat menggunakan *fingers_data.count(1)* dan kemudian mengubah nilai tersebut menjadi *string* menggunakan f-*string {totalFingers}* dalam perintah *cv2.putText()*. Langkah ini memastikan bahwa nilai jumlah jari yang ditampilkan di layar sesuai dengan format *string* yang diperlukan oleh fungsi *cv2.putText()* seperti pada Gambar 3.7.

```
54
55          totalFingers = fingers_data.count(1)
56          cv2.putText(img, f'Jumlah jari: {totalFingers}', (bbox[0], bbox[1] - 30),
57                      cv2.FONT_HERSHEY_PLAIN, 1.2, (0, 0, 255), 2)
58
```

Gambar 3.7 Konversi jumlah jari tangan ke *string*

Selain itu, ada juga proses konversi yang dilakukan untuk mengubah data jumlah jari yang terdeteksi menjadi *string* sebelum dikirimkan.

```
65          counter = sum(fingers_data_serial)
66          print("Nilai Counter: ", str(counter))
67          e = '\n'
68          send_message(str(counter))
69          send_message(e)
```

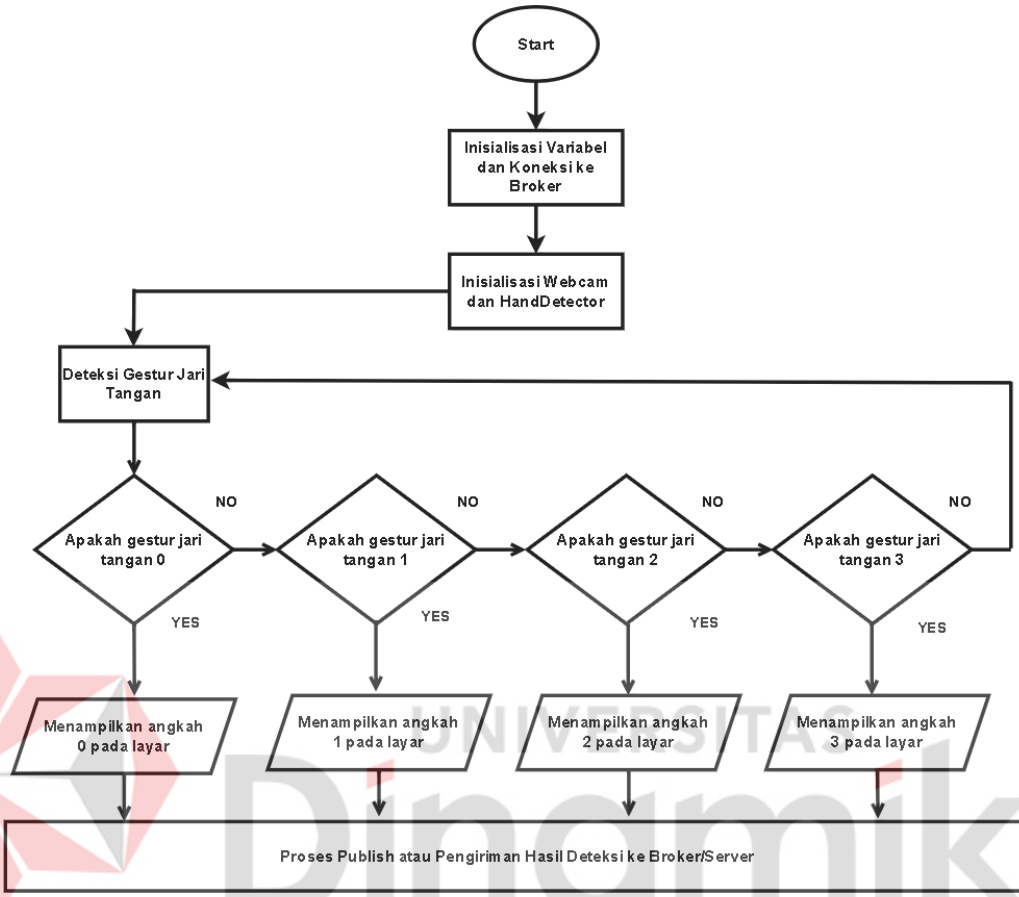
Gambar 3.8 Konversi nilai *counter* ke *string*

Adapun juga konversi nilai *counter* ke dalam bentuk string sebelum dikirim melalui MQTT *Broker* seperti yang dapat dilihat pada Gambar 3.8. Setelah nilai *counter* dihitung dengan menggunakan ***counter = sum(fingers_data_serial)***, nilai tersebut perlu diubah menjadi *string* menggunakan **str(counter)** sebelum dikirim ke MQTT *Broker* menggunakan fungsi ***send_message()***. Konversi ini diperlukan agar data yang dikirim ke MQTT *Broker* dapat diterima dan diinterpretasikan dengan benar oleh perangkat atau sistem penerima.

3.5 Program Perangkat Lunak

Program *computer vision* yang berjalan merupakan program yang sebelumnya telah buat oleh saudara peneliti Muhammad Aldi Fakhruddin dan ada beberapa penambaan yang penulis tambahkan untuk mencetak rata-rata FPS (*Frame per Second*) pada layar laptop serta konfigurasu ke MQTT *Broker*. Dilihat dari gambar 3.9, dapat dijelaskan bahwa program melakukan inisialisasi dan koneksi ke *broker* atau *server* terlebih dahulu. *Webcam* laptop melakukan deteksi bentuk gestur jari tangan apakah bentuk gestur jari tangan nol, satu, dua, atau tiga dan jikalau tidak ada jari tangan yang dideteksi maka program melakukan

looping atau perulangan ke proses deteksi gestur jari tangan untuk melakukan deteksi.

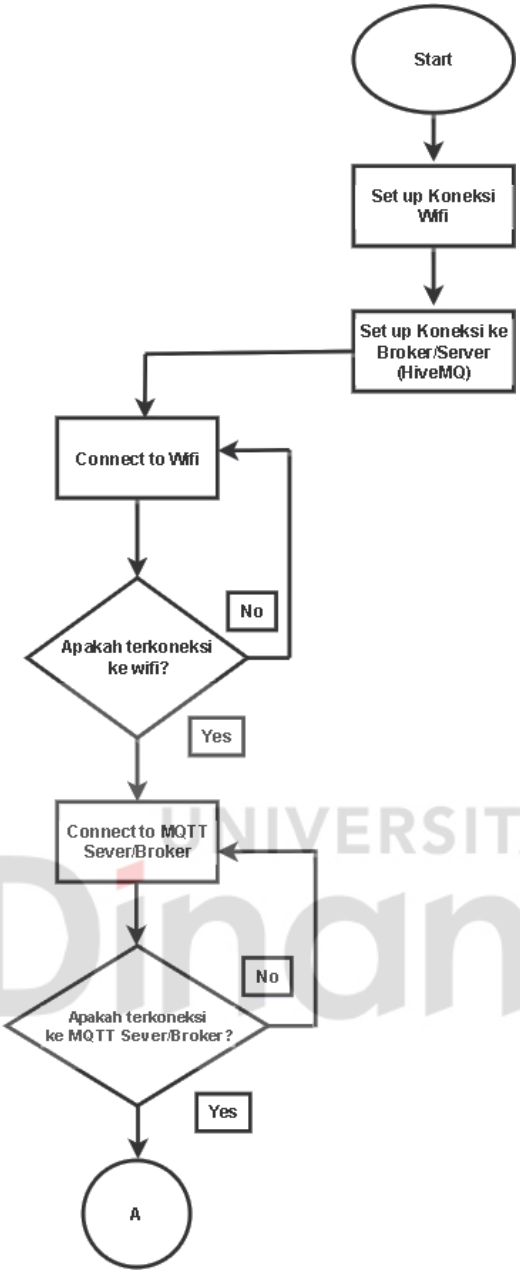


Gambar 3.9 *Flowchart* proses deteksi gestur jari tangan

Setelah memperoleh nilai yang dideteksi, dilanjutkan dengan menampilkan nilai tersebut ke layar laptop sebagai tanda kalau gestur jari tangan berhasil terdeteksi. Nilai atau hasil yang ditampikan pada layar laptop setelah itu dipublish ke MQTT *Broker* untuk selanjutnya diproses oleh *client* atau *subscriber*.

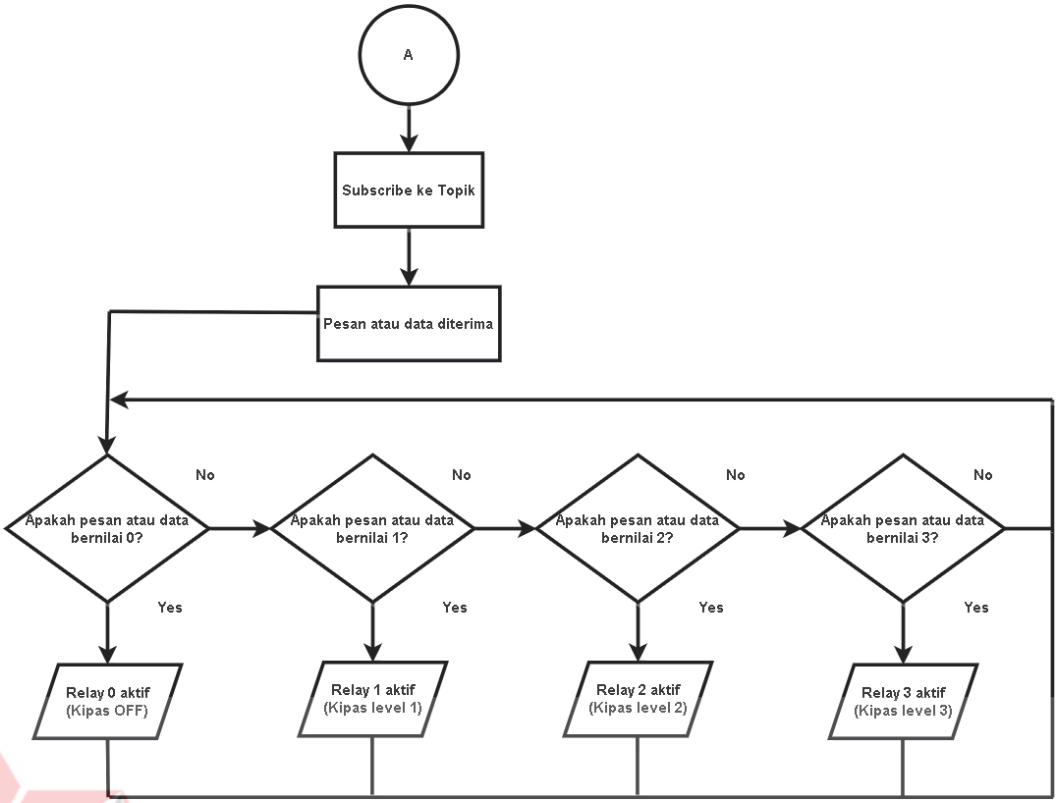
3.6 Flowchart Konfigurasi MQTT Broker

Flowchart program dibawah ini memungkinkan pengendalian perangkat melalui pesan MQTT yang dikirimkan ke topik "kipas_angin". Pesan-pesan tersebut diinterpretasikan oleh papan WeMos-D1R2 yang dibangun diatas ESP8266, dan relay akan diatur sesuai dengan pesan yang diterima.



Gambar 3.10 *Flowchart* konfigurasi *broker*

Setelah konfigurasi Wi-Fi dan *broker* berhasil, proses berikutnya dilanjutkan dengan berlangganan atau melakukan *subscribe* ke topik "kipas_angin", yang mana telah disepakati bersama untuk terjalinnya komunikasi antar *publisher, broker* dan *subscriber atau client*. Dengan melakukan *subscribe* atau berlangganan ke topik "kipas_angin", proses dilanjutkan dengan menerima pesan atau data dari *broker*.



Gambar 3.11 *Flowchart* program WeMos-D1R2

Ketika pesan diterima dari *broker*, program memproses *payload* pesan menjadi tipe data karakter. Selanjutnya, program memeriksa nilai karakter pesan untuk menentukan tindakan atau aksi yang harus dilakukan pada relay. Jika pesan adalah "0", semua relay dimatikan. Jika pesan adalah "1", Relay 1 dihidupkan dan relay lainnya dimatikan. Pesan "2" menghidupkan Relay 2 dan mematikan relay lainnya, sedangkan pesan "3" menghidupkan Relay 3 dan mematikan relay lainnya. Setiap tindakan relay dicetak ke *Serial Monitor* sebagai umpan balik. Program terus berjalan dalam *loop* utama untuk memeriksa koneksi MQTT dan memproses pesan yang diterima.

BAB IV
HASIL DAN PEMBAHASAN

Pada Bab 4 ini, terdapat pembahasan terkait pengujian dari sisi perangkat lunak dan perangkat keras. Dalam pengujian perangkat lunak, fokusnya adalah memastikan bahwa proses deteksi yang dilakukan oleh *computer vision* menggunakan *library* Medipipe berjalan sesuai hasil yang diharapkan hingga terkirimnya data deteksi gestur jari tangan ke MQTT *Broker*. Sementara itu, dalam pengujian perangkat keras, tujuannya adalah untuk menverifikasi kinerja alat beroperasi sesuai dengan yang telah ditentukan.

4.1 Pengujian Pengiriman dan Penerimaan Pesan antara Perangkat atau Sistem dengan MQTT Broker

4.1.1 Tujuan

Tujuan dari pengujian ini adalah untuk memastikan bahwa koneksi perangkat atau sistem ke MQTT *Broker* telah berhasil dan memeriksa proses pengiriman dan penerimaan pesan atau data hasil proses oleh *Computer Vision* menggunakan *library* Mediapipe dari *publisher* ke *subscriber* dapat berjalan dan terlaksana sesuai yang diinginkan, serta membandingkan seberapa cepat data yang diterima oleh *client* (*subscriber*).

4.1.2 Perlengkapan Yang Digunakan

Adapun alat-alat yang disiapkan sebelum melakukan percobaan:

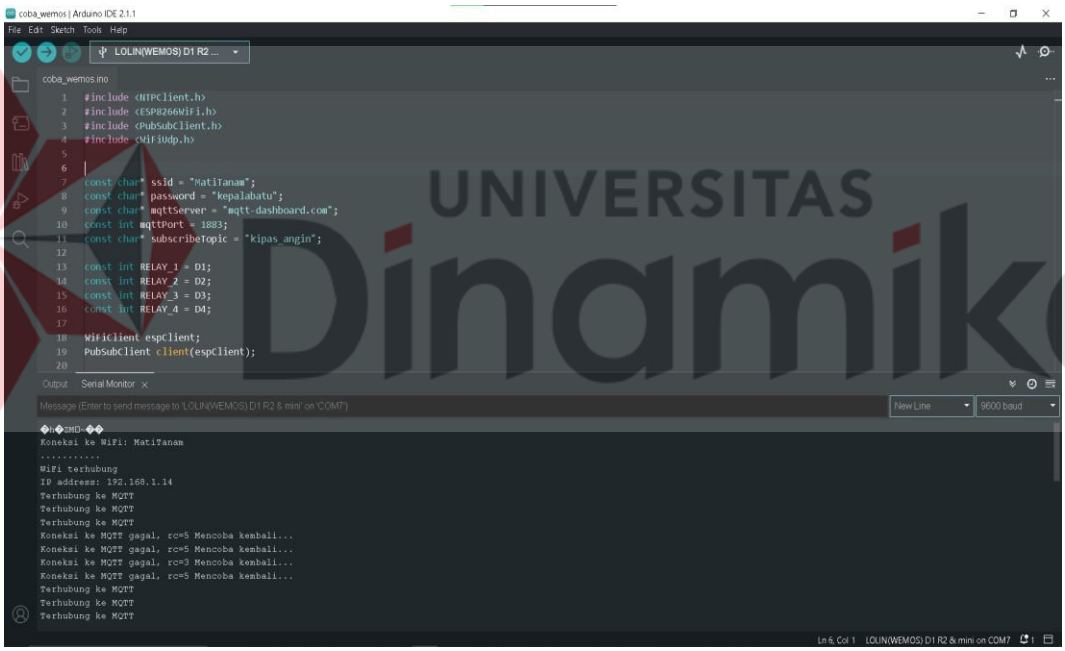
1. Modul WeMos-D1R2.
2. MQTT Websocket Client - HiveMQ
3. Laptop untuk eksekusi program *Computer Vision* dan program Arduino IDE.
4. Koneksi Wi-Fi.
5. Kipas Angin, 3 kecepatan (*low*, *medium*, *high*).
6. Modul Relay 4 *Channel.*

4.1.3 Cara Pengujian

Adapun langkah-langkah untuk melakukan pengujian ini adalah:

1. Menghubungkan Modul WeMos-D1R2 ke laptop untuk mengupload program dari Arduino IDE ke Modul WeMos-D1R2.
2. Memeriksa apakah Modul WeMos-D1R2 berhasil terkoneksi ke Wi-Fi dan *Broker*.
3. Menjalankan program *Computer Vision*.
4. Mengamati dan membandingkan data yang dikirim dari *publisher* ke *broker* dengan data yang diterima Modul WeMos-D1R2 di *Serial Monitor* pada *software* Arduino IDE.

4.1.4 Hasil Uji

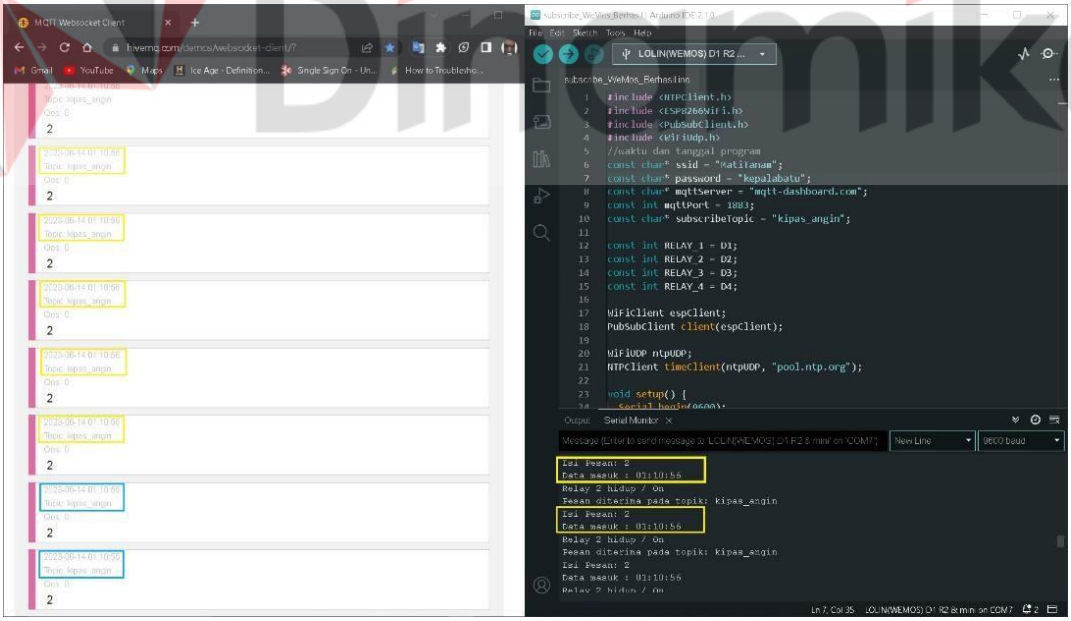


Gambar 4.1 Koneksi Wifi dan *Broker*

Pada Gambar 4.1, dapat dilihat bahwa Modul WeMos-D1R2 berhasil terhubung ke Wi-Fi dan juga terkoneksi ke MQTT *Broker* yang dituju. Dengan demikian, proses penerimaan dan pengontrolan data dari *broker* dan *client* (*subscriber*) dapat terjadi dan siap dieksekusi.

Gambar 4.2 Proses hasil deteksi oleh *computer vision*

Data yang diproses oleh *computer vision* dengan menerapkan metode Mediapipe kemudian diteruskan atau dikirimkan ke *broker* untuk selanjutnya diteruskan ke Modul WeMos-D1R2 sebagai *client* yang mengendalikan kipas angin. Output dari *computer vision* yang dikirimkan ke *broker* adalah data gestur jari tangan yang terdeteksi terangkat, seperti yang terlihat pada Gambar 4.2.



Gambar 4.3 Output *Serial Monitor* dan MQTT *Broker*

Berdasarkan hasil pengujian yang telah dilakukan seperti pada Gambar 4.3 dapat dikatakan, data yang diterima oleh Modul WeMos-D1R2 sudah sesuai dengan

data atau hasil yang diproses oleh *computer vision* 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.

Tabel 4.1 Pengujian seberapa *real-time* kontrol *level* kecepatan kipas angin

No	Broker HiveMQ (Data Gestur)	Waktu data masuk (Jm-Min-Dtk)	WeMos-D1R2 (Data gestur)	Waktu data masuk (Jm-Min-Dtk)	Status kipas		Keterangan
					On	Off	
1	1	11:19:54	1	19:19:54	√	-	Kipas level 1 (*high*)
2	2	11:20:01	-	-	-	-	Koneksi WeMos ke broker gagal
3	2	11:20:02	2	11:20:02	√		Kipas level 2 (*medium*)
4	1	11:20:08	1	11:20:08	√	-	Kipas level 1 (*high*)
5	3	11:20:14	-	-	-	-	Koneksi WeMos ke broker gagal
6	3	11:20:40	3	11:20:40	√	-	Kipas level 3 (*low*)
7	2	11:20:42	2	11:20:42	√	-	Kipas level 2 (medium)
8	3	11:20:54	3	11:20:54	√	-	Kipas level 3 (*low*)
9	0	11:20:55	0	11:20:55	-	√	Kipas level 0 (*off*/mati)
10	1	11:21:04	1	11:21:04	√	-	Kipas level 1 (*high*)
11	2	11:21:15	-	-	-	-	Koneksi WeMos ke broker gagal
12	2	11:21:20	-	-	-	-	Koneksi WeMos ke broker gagal
13	2	11:21:25	-	-	-	-	Koneksi WeMos ke broker gagal
14	2	11:21:46	-	-	-	-	Koneksi WeMos ke broker gagal
15	0	11:22:06	0	11:22:06	-	√	Kipas level 0 (*off*/mati)
16	2	11:22:11	2	11:22:11	√	-	Kipas level 2 (*medium*)
17	3	11:22:17	3	11:22:17	√	-	Kipas level 3 (*low*)
18	2	11:22:18	2	11:22:18	√	-	Kipas level 2 (*medium*)
19	1	11:22:20	-	-	-	-	Koneksi WeMos ke broker gagal

No	Broker HiveMQ (Data Gestur)	Waktu data masuk (Jm-Min-Dtk)	WeMos-D1R2 (Data gestur)	Waktu data masuk (Jm-Min-Dtk)	Status kipas		Keterangan
					On	Off	
20	1	11:22:32	1	11:22:32	√		Kipas level 1 (*high*)
21	3	11:22:32	3	11:22:32	√	-	Kipas level 3 (*low*)
22	3	11:22:36	3	11:22:36	√	-	Kipas level 3 (*low*)
23	3	11:22:37	3	11:22:37	√	-	Kipas level 3 (*low*)
24	0	11:22:40	-	-	-	-	Koneksi WeMos ke broker gagal
25	0	11:22:47	0	11:22:47	-	√	Kipas level 0 (*off*/mati)
26	0	11:22:48	0	11:22:48	-	√	Kipas level 0 (*off*/mati)
27	1	11:22:57	1	11:22:57	√	-	Kipas level 1 (*high*)
28	2	11:23:02	2	11:23:02	√	-	Kipas level 2 (*medium*)
29	3	11:23:17	2	11:23:17	√	-	Kipas level 3 (*low*)
30	0	11:23:20	-	-	-	-	Koneksi WeMos ke broker gagal
Total pengontrolan kipas angin yang berhasil					**21**		
Total pengontrolan kipas angin yang tidak berhasil					**9**		

Berdasarkan Tabel 4.1, telah dilakukan pengujian untuk mengetahui seberapa *real-time* proses pengontrolan kecepatan putaran kipas angin melalui deteksi bentuk gestur jari tangan secara jarak jauh. Pengujian dilakukan sebanyak 30 kali percobaan dengan tujuan untuk membandingkan waktu data yang berhasil terkirim ke *broker* dan waktu data yang diterima atau masuk ke WeMos-D1R2, serta sekaligus mencatat hasil atau output dari kipas angin. Sehingga didapatkan proses pengontrolan yang berhasil dilakukan sebanyak 21 dengan total pengontrolan kipas angin yang tidak berhasil berjumlah 9. Gagalnya pengontrolan sebanyak 9 kali dari 30 kali pengambilan data ini, dikarenakan WeMos-D1R2 gagal melakukan koneksi ke broker.

Waktu penerimaan pesan atau data oleh WeMos-D1R2 seperti yang dapat dilihat pada Tabel 4.1, dapat dikatakan cukup *real-time*. Hal ini dibutkikan dengan

cara membandikan waktu data yang masuk di broker dengan waktu penerimaan data pada sisi WeMos-D1R2.

4.2 Pengontrolan Level Kecepatan Putaran Kipas Angin secara Computer Vision dari Jarak Jauh Menggunakan Protokol MQTT

4.2.1 Tujuan

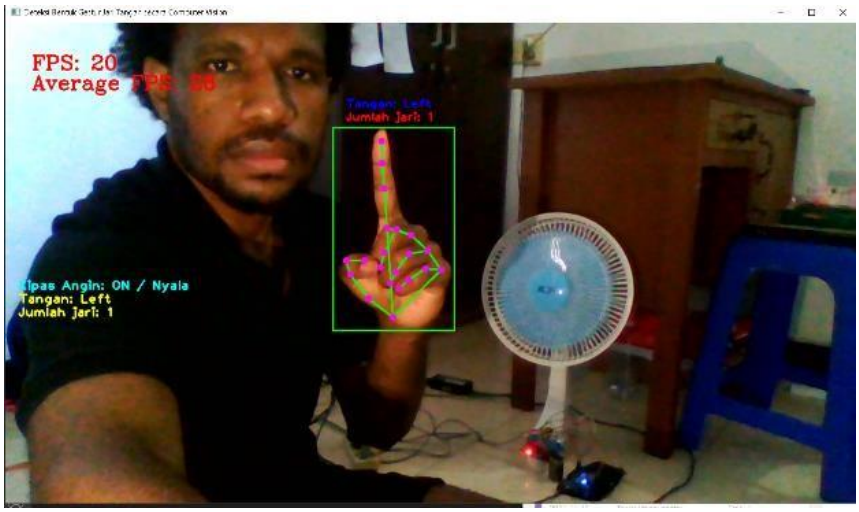
Tujuannya adalah mengontrol *level* kecepatan putaran kipas angin secara jarak jauh melalui *computer vision*. Percobaan ini juga bertujuan untuk memeriksa apakah *output* atau keluaran kipas angin sesuai dengan inputan yang diberikan oleh *computer vision* melalui *MQTT Broker*.

4.2.2 Prosedur Pengontrolan Level Kecepatan Putaran Kipas Angin secara Computer Vision dari Jarak Jauh Menggunakan Protokol MQTT

Dalam melakukan pengujian, adapun langkah-langkahnya:

1. Mengirimkan inputan bentuk gesturs jari tangan pada *computer vision*.
2. Memulai proses deteksi bentuk gestur jari tangan dan proses penerimaan oleh MQTT *Broker*.
3. Memastikan kipas dan sistem perangkat keras yang telah dibangun dalam keadaan *On* (hidup).

4.2.3 Hasil Uji Pengontrolan Level Kecepatan Putaran Kipas Angin secara Jarak Jauh melalui Deteksi Bentuk Gestur Jari Tangan oleh Computer Vision



Gambar 4.4 Output pengontrolan *level* putaran kipas angin

Berdasarkan hasil pengujian pada Gambar 4.4 yaitu pengontrolan *level* kecepatan putaran kipas angin berbasis IoT, sudah berjalan sesuai harapan peneliti. Putaran kecepatan kipas angin meningkat ke *level* kecepatan 1 (*high*) berdasarkan data yang diterima melalui MQTT *Broker*.

Tabel 4.2 Pengujian putaran kipas angin melalui *computer vision*

No	Nama	Jarak (cm)	Gestur	Status level Kecepatan Kipas				Hasil deteksi		FPS	Keterangan
				0	1	2	3	Benar	Salah		
1	Fredi	100	0	√				√	-	22	Berhasil (kipas Off)
			1		√			√	-	24	Berhasil (kipas On)
			2			√		√	-	21	Berhasil (kipas On)
			3				√	√	-	23	Berhasil (kipas On)
		150	0	√				√	-	31	Berhasil (kipas Off)
			1		√			√	-	31	Berhasil (kipas On)
			2			√		√	-	31	Berhasil (kipas On)
			3				√	√	-	31	Kipas On setelah delay yang panjang

No	Nama	Jarak (cm)	Gestur	Status level Kecepatan Kipas				Hasil deteksi		FPS	Keterangan
				0	1	2	3	Benar	Salah		
		200	0	√				√	-	31	Kipas Off setelah delay yang panjang
			1		√			√	-	31	Kipas On setelah delay yang panjang
			2			√		√	-	31	Berhasil (kipas On)
			3				√	√	-	31	Berhasil (kipas On)
2	Yosea	100	0	√				√	-	31	Kipas Off setelah delay yang panjang
			1		√			√	-	31	Kipas On setelah delay yang panjang
			2			√		√	-	31	Kipas On setelah delay yang panjang
			3				√	√	-	31	Kipas On setelah delay yang panjang
		150	0	√				√	-	31	Kipas Off setelah delay yang panjang
			1		√			√	-	32	Berhasil (kipas On)
			2			√		√	-	31	Kipas On setelah delay yang panjang
			3				√	√	-	31	Kipas On setelah delay yang panjang
		200	0	√				√	-	31	Kipas Off setelah delay yang panjang
			1		√			√	-	31	Kipas On setelah delay yang panjang
			2			√		√	-	31	Kipas On setelah delay yang panjang
			3				√	√	-	31	Berhasil (kipas On)
3	Niol	100	0	√				√	-	31	Kipas Off setelah delay yang panjang
			1		√			√	-	31	Kipas On setelah delay yang panjang
			2			√		√	-	31	Berhasil (kipas On)

No	Nama	Jarak (cm)	Gestur	Status level Kecepatan Kipas				Hasil deteksi		FPS	Keterangan
				0	1	2	3	Benar	Salah		
			3				√	√	-	31	Kipas On setelah delay yang panjang
		150	0	√				√	-	31	Kipas Off setelah delay yang panjang
			1		√			√	-	31	Berhasil (kipas On)
			2			√		√	-	31	Berhasil (kipas On)
			3				√	√	-	31	Kipas On setelah delay yang panjang
		200	0	√				√	-	31	Berhasil (kipas Off)
			1		√			√	-	31	Kipas On setelah delay yang panjang
			2			√		√	-	31	Kipas On setelah delay yang panjang
			3				√	√	-	31	Berhasil (kipas On)
4	Dani	100	0	√					-	31	Kipas Off setelah delay yang panjang
			1		√			√	-	31	Kipas On setelah delay yang panjang
			2			√		√	-	31	Kipas On setelah delay yang panjang
			3				√	√	-	31	Kipas On setelah delay yang panjang
		150	0	√				√	-	31	Kipas Off setelah delay yang panjang
			1		√			√	-	31	Berhasil (kipas On)
			2			√		√	-	31	Berhasil (kipas On)
			3				√	√	-	31	Kipas On setelah delay yang panjang
		200	0	√				√	-	31	Berhasil (kipas Off)
			1		√			√	-	31	Kipas On setelah delay yang panjang
			2			√		√	-	31	Kipas On setelah delay yang panjang

No	Nama	Jarak (cm)	Gestur	Status level Kecepatan Kipas				Hasil deteksi		FPS	Keterangan
				0	1	2	3	Benar	Salah		
			3				√	√	-	31	Kipas On setelah delay yang panjang
5	Abet	100	0	√				√	-	31	Berhasil (kipas Off)
			1		√			√	-	31	Berhasil (kipas On)
			2			√		√	-	31	Kipas On setelah delay yang panjang
			3				√	√	-	31	Berhasil (kipas On)
		150	0	√				√	-	31	Berhasil (kipas Off)
			1		√			√	-	31	Berhasil (kipas On)
			2			√		√	-	31	Kipas On setelah delay yang panjang
			3				√	√	-	31	Berhasil (kipas On)
		200	0	√				√	-	31	Kipas Off setelah delay yang panjang
			1		√			√	-	31	Kipas On setelah delay yang panjang
			2			√		√	-	31	Kipas On setelah delay yang panjang
			3				√	√	-	31	Kipas On setelah delay yang panjang
Total gestur terdeteksi								**60**	**0**		

Pengujian yang telah dilakukan seperti pada Tabel 4.2 merupakan hasil pengontrolan *level* kecepatan putaran kipas angin melalui proses deteksi bentuk gestur jari tangan menggunakan *computer vision* secara jarak jauh. Proses untuk mengontrol *level* kecepatan kipas angin dilakukan terhadap 5 subjek yang berbeda dengan masing-masing subjek diberikan 3 jarak yaitu, jarak 100cm, 150cm dan jarak 200cm. Output dari setiap gestur jari tangan yang didapatkan akan menjadi inputan bagi Modul WeMos-D1R2 untuk mengontrol *level* kecepatan putaran kipas angin.

Dapat dilihat hasil uji dari setiap subjek dan jarak yang berbeda untuk 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.

Namun, dari sisi proses menongtrol *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* gestru 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*. Penyebab terjadinya jeda atau delay yang berkepanjangan ini disebabkan oleh padatnya traffic jaringan pada saat siang hari, dimana banyaknya *device* dan *user* yang juga terhubung ke jaringan wifi yang sama, sehingga menyebabkan koneksi internet atau Wi-Fi pada WeMos-D1R2 tidak stabil dan terus melakukan reconnect atau koneksi ulang ke broker ketika kembali terhubung ke jaringan wifi. Selain itu juga, penentuan waktu pengambilan data deteksi gestru jari tangan dapat menyebabkan proses kontrol *level* kecepatan kipas angin terjadi jeda atau delay.

Secara keseluruhan proses kontrol *level* kecepatan putaran kipas angin secara jarak jauh menggunakan *computer vision* dapat dilakukan. Hal ini dapat dibuktikan dari setiap percobaan yang dapat dilihat pada Tabel 4.2, yang dilakukan oleh setiap subjek yang berbeda.

Meskipun demikian, percobaan yang dilakukan oleh subjek Fredi memiliki nilai FPS yang rendah. Keberhasilan kontrol *level* kecepatan kipas angin yang dilakukan subjek Fredi dikarenakan proses pengambilan data dilakukan ditengah malam, sehingga pengguna atau *user* lain yang terhubung ke jaringan wifi yang sama lebih sedikit. Namun, nilai FPS (*Frame per Second*) dari percobaan ini terbilang rendah dari percobaan yang dilakukan subjek lain disiang hari.

4.3 Pengujian Tingkat Akurasi Dalam Proses Pengontrolan Kipas Angin Melalui Deteksi Bentuk Gestru Jari Tangan Secara Jarak Jauh

4.3.1 Tujuan

Tujuan dari pengujian ini adalah untuk mencari berapa besar tingkat akurasi proses pengontrolan level kecepatan putaran kipas angin yang dilakukan secara jarak jauh berbasi computer visiom.

4.3.2 Hasil Pengujian Tingkat Proses Pengontrolan Kipas Angin Secara Jarak Jauh Mengunakan Library Mediapipe

Berdasarkan pada Tabel 4.3, 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.

Tabel 4.3 Pengujian akurasi gestur jari tangan

No	Nama	Jarak (cm)	Gestur (10 kali percobaan)	Hasil deteksi		Akurasi (%)	FPS	Keterangan
				Benar	Salah			
1	Fredi	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	

No	Nama	Jarak (cm)	Gestur (10 kali percobaan)	Hasil deteksi		Akurasi (%)	FPS	Keterangan
				Benar	Salah			
			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	
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 2						**94**	**29.9**	
Rata-rata gestur 2						**99**	**28.6**	
Rata-rata gestur 3						**99**	**31.4**	

Dari pengujian yang dilakukan untuk mendapatkan tingkat akurasi yang tinggi pada Tabel 4.3, 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.

BAB V
PENUTUP

5.1 Kesimpulan

Beberapa kesimpulan dapat diambil dari hasil Tugas Akhir ini, yaitu:

1. Pengontrolan kipas angin secara jarak jauh melalui deteksi bentuk gestur jari tangan secara *computer vison* berbasis IoT dengan protokol MQTT untuk mengontrol *level* kecepatan putaran kipas angin dapat dilakukan secara *real-time*.
2. Hasil percobaan untuk mencari akurasi gestur jari tangan dan FPS pada jarak 100 cm didapatkan sebagai berikut:
   a) Gestur 0 memiliki akurasi sebesar 100% dengan FPS sebesar 28.6.
   b) Gestur 1 memiliki akurasi sebesar 94% dengan FPS sebesar 29.9.
   c) Gestur 2 memiliki akurasi sebesar 99% dengan FPS sebesar 28.6.
   d) Gestur 3 memiliki akurasi sebesar 99% dengan FPS sebesar 31.4.
3. Hasil pengujian perubahan pada jarak 100 cm untuk deteksi bentuk gestur jari tangan 0, 1, 2, dan 3 dari lima orang mempunyai akurasi 100%.
4. Hasil pengujian perubahan pada jarak 150 cm untuk deteksi bentuk gestur jari tangan 0, 1, 2, dan 3 dari lima orang mempunyai akurasi 100%.
5. Hasil pengujian perubahan pada jarak 200 cm pada lima orang untuk proses deteksi bentuk gestur jari tangan 0 akurasinya 80%, gestur jari tangan 1 akurasinya 80%, sedangkan gestur jari tangan 2 dan 3 akurasinya 100%.
6. Hasil pengujian perubahan pada jarak lebih dari 200 cm pada lima orang untuk proses deteksi bentuk gestur jari tangan 0 akurasinya 80%, gestur jari tangan 1 akurasinya 80%, gestur jari tangan 2 akurasinya 100%, sedangkan gestur jari tangan 3 akurasinya 90%.

5.2 Saran

Saran-saran yang dapat diberikan untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Pengujian deteksi bentuk gestur jari tangan menggunakan *library* Mediapipe dapat dibilang cukup akurat hanya saja diperlukan spesifikasi perangkat keras

yang cukup tinggi untuk dijalankan diatasnya agar proses *computer vision* dapat berjalan dengan cepat.

2. Pemilihan mikrokontroler yang mendukung komunikasi secara *wireless* dan *real-time*. Namun, perluh dipertimbangkan seperti program yang dapat dibilang besar yang menggunakan banyak *library*, *variable global* yang banyak dapat menyebabkan mikrokontroler itu cepat panas dan rusak.
3. Pemilihan MQTT *Broker* yang menyediakan banyak *fitur dan dashboard* untuk pengelolahan data masukan serta pengamanan yang kuat agar data yang dikirim tidak hilang. Namun, hal itu dapat disesuaikan dengan anggaran yang dimiliki karena *server* atau *broker* yang menyediakan semua itu hampir semuanya berbayar.

DAFTAR PUSTAKA

ANDY. (2019, 12). *Wemos D1 R2 Wifi Arduino Development Board*. From Mengenal Teknologi Sistem Informasi Komputer: https://andydharmalau.com/wemos-d1-r2-wifi-arduino-development-board/

BINUS UNIVERSITY. (2022, Juli 27). From student-activity.binus.ac.id: https://student-activity.binus.ac.id/himtek/2022/07/27/esp32/

erintafifah. (2021, Oktober 8). *KMTek*. From kmtech: https://www.kmtech.id/post/mengenal-perangkat-lunak-arduino-ide

Fakhruddin, M. A. (2023). Sistem Deteksi Gestur Jari Tangan Menggunakan Mediapipe Dan Faster-Rcnn Untuk Mengontrol Kecepatan Kipas Angin.

Hithaishi, R. (2022). Optimized Hand Gesture Based Home Automation For Feebles. *International Journal of Research in Engineering and Science (IJRES), X*.

Kukil. (2022, March 1). *Introduction to MediaPipe*. From learnopencv.com: https://learnopencv.com/introduction-to-mediapipe/

Lubis, M. Y. (2021). Implementasi Artificial Intelligence Pada System Manufaktur Terpadu. *Seminar Nasional Teknik (SEMNASTEK) UISU*, 1-2.

Rahman, B., & Imelda. (2020, April 20). Prototipe Sistem Kontrol Smart HomeBerbasis IoTDengan Metode MQTT Menggunakan Google Asisstant. *JURNAL RESTI (Rekayas a Sistem dan T eknol ogi Informasi), IV*, 303-310. From http://jurnal.iaii.or.id/index.php/RESTI/article/view/1721

Razor, A. (2021, Maret 5). *ALDYRAZOR.COM*. From aldyrazor: https://www.aldyrazor.com/2020/05/modul-relay-arduino.html

susanto, F., Prasiani, N. K., & Darmawan, P. (2022). Implementasi Internet Of Things Dalam Kehidupan Sehari-Hari. *JURNAL IMAGINE, II*, 35-40. doi:https://doi.org/10.35886/imagine.v2i1.329

Szeliski, R. (2022). *Computer Vision: Algorithms and Applications*. Springer Nature.