

PENGENDALI LAMPU MENGGUNAKAN GESTUR TANGAN BERBASIS MEDIAPIPE DENGAN JAVASCRIPT

Emmanuel Manggala Nusa, Dian Widiyanto Chandra

Teknik Informatika, Universitas Kristen Satya Wacana
Jalan Diponegoro No. 52-60, Salatiga, Jawa Tengah 50711, Indonesia
e.manggalanusa@gmail.com

ABSTRAK

Perkembangan teknologi interaksi tanpa sentuh, khususnya pengenalan gestur tangan, menawarkan cara kendali perangkat yang lebih alami dan efisien, namun penerapannya pada perangkat IoT masih menghadapi tantangan terkait stabilitas deteksi dalam berbagai kondisi lingkungan. Penelitian ini bertujuan merancang dan mengimplementasikan sistem kendali lampu berbasis *web* yang mampu mendeteksi gestur tangan secara *real-time* menggunakan MediaPipe sebagai pendeteksi *landmark* dan ESP32 sebagai aktuator kendali perangkat. Sistem dikembangkan dalam arsitektur *client-server*, di mana aplikasi *web* memproses citra kamera dan mengirimkan perintah *ON/OFF* ke ESP32 melalui protokol HTTP. Pengujian dilakukan pada ruangan tertutup dengan dua kondisi pencahayaan lampu 3 Watt dan 10 Watt serta variasi jarak deteksi 30–290 cm. Hasil menunjukkan bahwa sistem mencapai akurasi 100% pada seluruh rentang jarak ketika pencahayaan mencukupi, sedangkan penurunan akurasi terjadi pada kondisi cahaya sangat rendah yaitu 5 lux dengan akurasi 97% untuk gestur *ON* dan 80% untuk gestur *OFF* pada jarak 290 cm. Temuan ini menegaskan bahwa pencahayaan merupakan faktor utama yang memengaruhi keberhasilan deteksi gestur dalam sistem ini. Penelitian ini berkontribusi pada pengembangan sistem kendali perangkat IoT berbasis gestur tangan yang dapat dijalankan sepenuhnya melalui antarmuka *web* tanpa memerlukan aplikasi tambahan.

Kata kunci : *Gesture Recognition, MediaPipe, ESP32, Internet of Things, Kendali Lampu, Web-Based Control*

1. PENDAHULUAN

Perkembangan teknologi pada era Industri 4.0 mendorong integrasi yang semakin erat antara manusia dan perangkat digital, khususnya melalui konsep *Human-Computer Interaction* (HCI), *Internet of Things* (IoT), dan kecerdasan buatan. Salah satu bentuk interaksi yang berkembang pesat adalah pengenalan gestur tangan (*hand gesture recognition*), yang memungkinkan pengguna mengendalikan perangkat tanpa kontak fisik sehingga interaksi menjadi lebih alami dan intuitif [1], [2], [3]. Teknologi ini relevan diterapkan pada sistem rumah pintar (*smart home*), terutama dalam konteks efisiensi dan kenyamanan pengguna [4], [5], [6], [7], [8].

MediaPipe, sebagai *framework* pemrosesan citra *real-time*, menyediakan kemampuan deteksi dan pelacakan tangan melalui 21 titik *landmark* sehingga memungkinkan pengenalan gestur secara stabil. Meskipun teknologi ini telah banyak digunakan dalam aplikasi berbasis perangkat lunak seperti presentasi atau sistem interaktif berbasis layar, penerapannya untuk mengendalikan perangkat keras IoT secara langsung masih relatif terbatas [1], [2], [4], [9], [10], [11], [12], [13], [14], [15]. Di sisi lain, mikrokontroler ESP32 memungkinkan komunikasi nirkabel yang efisien dan dapat berfungsi sebagai pengendali perangkat elektronik melalui jaringan lokal. Integrasi MediaPipe pada antarmuka *web* berbasis JavaScript dengan ESP32 berpotensi menghasilkan sistem kendali yang fleksibel, mudah diakses, dan tidak memerlukan aplikasi atau peralatan tambahan.

Penelitian ini mengembangkan sistem kendali lampu berbasis gestur tangan yang memanfaatkan

MediaPipe untuk deteksi visual dan ESP32 sebagai aktuator. Sistem mengenali dua gestur sederhana, yaitu kondisi *ON* ketika seluruh jari terbuka dan *OFF* ketika kondisi jari tidak sepenuhnya terbuka. Untuk mengevaluasi performa sistem, dilakukan pengujian pada ruangan tertutup dengan dua tingkat pencahayaan berbeda yaitu 10 Watt dan 3 Watt dan variasi jarak 30–290 cm. Pengujian ini bertujuan untuk menilai keandalan deteksi gestur dalam kondisi lingkungan yang berbeda.

Secara khusus, penelitian ini bertujuan: (1) merancang dan mengimplementasikan sistem kendali lampu berbasis *web* yang mampu mendeteksi gestur tangan secara *real-time* menggunakan MediaPipe dan ESP32, serta (2) menganalisis tingkat akurasi sistem dalam mengenali gestur tangan pada berbagai kondisi pencahayaan dan jarak deteksi. Adapun hipotesis penelitian ini adalah bahwa sistem akan mampu mencapai tingkat akurasi di atas 90% selama pencahayaan berada pada tingkat yang mencukupi, sementara performa diprediksi menurun pada intensitas cahaya yang rendah. Dengan demikian, penelitian ini diharapkan memberikan kontribusi terhadap pengembangan teknologi interaksi tanpa sentuh dalam aplikasi IoT, sekaligus menyediakan solusi alternatif yang lebih inklusif bagi pengguna dengan keterbatasan mobilitas atau kondisi tertentu yang menyulitkan interaksi fisik dengan perangkat.

2. TINJAUAN PUSTAKA

Pengenalan gestur tangan merupakan salah satu bentuk interaksi dalam *Human-Computer Interaction* (HCI) yang memungkinkan pengguna berkomunikasi

dengan sistem tanpa kontak fisik [2], [3]. MediaPipe menjadi salah satu *framework* yang banyak digunakan dalam pengembangan teknologi ini karena kemampuannya mendeteksi dan melacak 21 titik *landmark* pada tangan secara *real-time* dengan stabilitas yang baik. Sejumlah penelitian telah memanfaatkan MediaPipe dalam konteks kendali perangkat lunak. Agustiani dkk. serta Khairianto dan Firdaus menerapkan deteksi gestur tangan untuk mengontrol transisi slide pada presentasi PowerPoint, di mana gestur seperti tangan terbuka, menunjuk, atau mengepal diinterpretasikan sebagai perintah navigasi [12], [14]. Chalik dkk. juga mengembangkan *virtual mouse* yang dikendalikan melalui gerakan tangan menggunakan integrasi MediaPipe dan OpenCV, menunjukkan bahwa MediaPipe mampu menyediakan informasi koordinat yang konsisten untuk pengendalian kursor [15].

Selain untuk pengendalian perangkat lunak, integrasi MediaPipe dengan perangkat IoT juga menjadi perhatian dalam penelitian terkini. Yang dkk. merancang sistem kendali lampu rumah pintar menggunakan MediaPipe dan mikrokontroler ESP8266 dengan hasil akurasi pengenalan gestur angka mendekati 98% [8]. Fakhruddin dkk. mengembangkan sistem pengaturan kecepatan kipas berbasis Arduino dengan perintah yang dihasilkan dari kombinasi MediaPipe dan Faster-RCNN, memperkuat bukti bahwa MediaPipe mampu menjadi modul deteksi visual dalam sistem berbasis mikrokontroler [11]. Penelitian lain oleh Nautica menunjukkan bahwa MediaPipe lebih efisien dibandingkan model CNN konvensional dalam aplikasi media pembelajaran berbasis penghitungan jari bagi anak usia dini [13]. Sementara itu, Samsinar dkk. memanfaatkan MediaPipe untuk mendeteksi isyarat tertentu sebagai bagian dari sistem keamanan CCTV cerdas yang mengirim notifikasi otomatis [5].

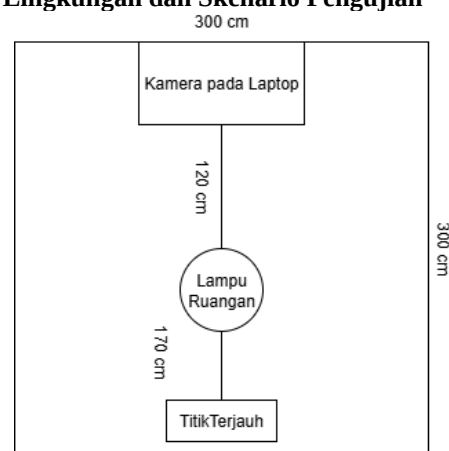
Berdasarkan kajian tersebut, MediaPipe terbukti sebagai solusi yang efektif untuk pengenalan gestur tangan baik dalam konteks perangkat lunak maupun perangkat keras IoT. Namun, sebagian besar studi IoT sebelumnya masih menggunakan aplikasi *desktop* atau *mobile* sebagai antarmuka pemrosesan gestur, bukan pemrosesan langsung melalui *web*. Dengan demikian, berbeda dari penelitian sebelumnya yang berfokus pada pemrosesan berbasis aplikasi, penelitian ini menekankan integrasi *gesture recognition* berbasis *web* menggunakan MediaPipe dengan perangkat keras IoT (ESP32) sebagai aktuator, yang hingga kini masih jarang dieksplorasi. Pendekatan ini menawarkan keunggulan berupa aksesibilitas lebih tinggi, fleksibilitas pemakaian, serta pengurangan kebutuhan instalasi aplikasi tambahan, sehingga relevan untuk implementasi IoT modern berbasis *web*.

3. METODE PENELITIAN

Penelitian ini mengembangkan sistem kendali lampu berbasis gestur tangan dengan memanfaatkan MediaPipe sebagai pendeteksi *landmark* tangan pada

antarmuka *web* berbasis ReactJS, serta mikrokontroler ESP32 sebagai pengendali lampu LED. Sistem diimplementasikan dalam arsitektur *client-server*, di mana aplikasi *web* bertindak sebagai *client* untuk memproses citra secara *real-time*, sedangkan ESP32 berfungsi sebagai *server* yang menerima perintah dan menjalankan aktuasi pada perangkat. Penelitian ini dibatasi pada deteksi gestur *ON* dan *OFF* menggunakan satu tangan, dengan lingkungan pengujian pada ruangan tertutup dan dua kondisi pencahayaan. Pengujian tidak mencakup variasi bentuk tangan, perbedaan warna kulit, maupun gestur kompleks, sehingga hasil yang diperoleh berfokus pada performa sistem dalam kondisi terkontrol sesuai ruang lingkup penelitian.

3.1. Lingkungan dan Skenario Pengujian



Gambar 1. Denah ruang pengujian

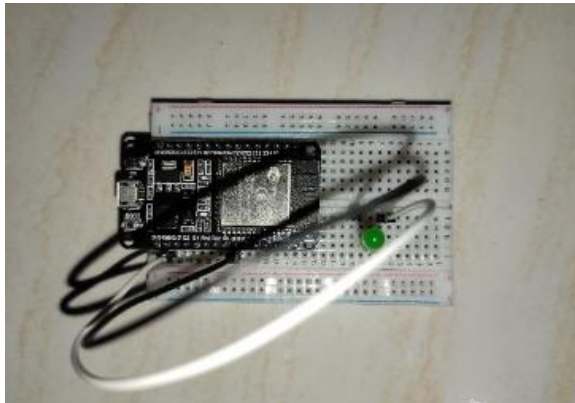
Pengujian dilakukan pada ruangan tertutup berukuran 300×300 cm pada malam hari untuk menjaga konsistensi intensitas cahaya. Kamera laptop ditempatkan pada posisi tetap dan menghadap area pengujian dengan jarak variasi antara 30 cm hingga 290 cm. Sumber pencahayaan menggunakan lampu LED 10 Watt dan 3 Watt untuk membentuk dua pengujian dengan kondisi pencahayaan yang berbeda. Pengaturan ini bertujuan untuk meminimalkan variabel luar yang dapat memengaruhi kualitas citra dan akurasi deteksi.

3.2. Perangkat Penelitian

Perangkat penelitian yang digunakan terdiri atas perangkat keras dan perangkat lunak yang mendukung proses deteksi gestur serta pengendalian perangkat IoT secara *real-time*. Laptop Asus TUF F15 digunakan sebagai *platform* utama untuk menjalankan aplikasi *web* dan pemrosesan citra, dengan spesifikasi prosesor 11th Gen Intel® Core™ i7-11800H 2.30 GHz, RAM 16 GB, serta sistem operasi Windows 11 Home Single Language 64-bit. Kamera bawaan laptop beresolusi 720p digunakan sebagai sensor akuisisi citra dengan kemampuan perekaman 30 fps. Mikrokontroler ESP32 berfungsi sebagai unit kendali perangkat IoT dan menjadi server penerima perintah *ON/OFF* dari aplikasi *web* melalui protokol HTTP. Lampu LED 5

mm digunakan sebagai aktuator yang merepresentasikan perangkat yang dikendalikan.

Perangkat lunak yang digunakan meliputi Google Chrome sebagai lingkungan eksekusi aplikasi, Visual Studio Code sebagai *editor* pengembangan antarmuka *web* berbasis ReactJS, Node.js sebagai *runtime*, serta Arduino IDE untuk pemrograman ESP32. Sistem deteksi gestur memanfaatkan *library* MediaPipe Hands yang melakukan ekstraksi 21 titik *landmark* tangan secara *real-time*. Untuk memastikan intensitas cahaya pada setiap pengujian, aplikasi Lux Light Meter Pro digunakan sebagai alat ukur.



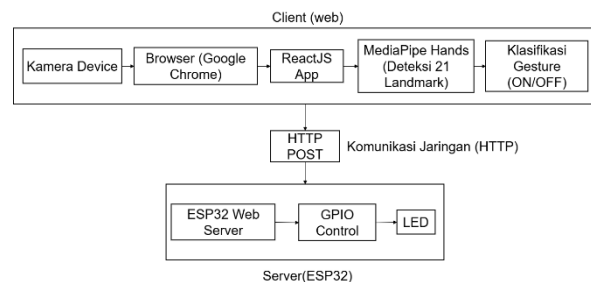
Gambar 2. Rangkaian ESP32

Rangkaian perangkat keras ditunjukkan pada Gambar 2. Mikrokontroler ESP32 dipasang pada breadboard dan dihubungkan dengan lampu LED sebagai aktuator *output*. Kaki anoda LED terhubung ke pin GPIO 18, sedangkan kaki katoda dihubungkan ke *ground* (GND) untuk menjaga referensi tegangan yang stabil. Rangkaian memperoleh suplai daya langsung melalui *port* USB laptop sehingga tidak memerlukan sumber daya eksternal tambahan. Konfigurasi ini memungkinkan LED merespons perintah logika digital HIGH dan LOW yang dikirimkan oleh aplikasi *web* melalui protokol HTTP untuk menghidupkan atau mematikan LED. Rangkaian ini dirancang sederhana dan modular sehingga mempermudah proses pengujian, pemantauan respons sistem, serta memudahkan replikasi pada penelitian selanjutnya.

3.3. Alur Kerja Sistem

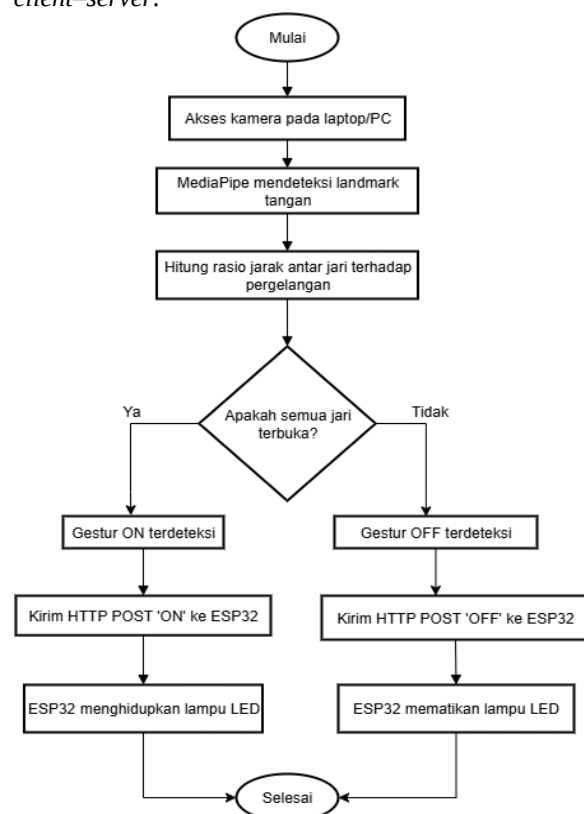
Sistem kendali lampu berbasis gestur tangan ini bekerja dengan mengintegrasikan aplikasi *web* sebagai pendeteksi gestur dan mikrokontroler ESP32 sebagai aktuator. Kamera pada laptop menangkap citra tangan pengguna, kemudian MediaPipe memproses citra tersebut untuk mengekstraksi 21 titik *landmark* tangan secara *real-time*. Berdasarkan informasi koordinat *landmark*, sistem melakukan klasifikasi gestur menggunakan perhitungan rasio panjang relatif antar ruas jari untuk menentukan apakah gestur yang ditampilkan merupakan kondisi ON atau OFF. Setelah gestur berhasil dikenali, aplikasi *web* mengirimkan perintah ke ESP32 melalui protokol HTTP, dan ESP32 mengeksekusi perubahan status pada lampu LED.

Dengan demikian, seluruh proses mulai dari deteksi visual hingga kendali perangkat IoT berlangsung secara *real-time* dalam arsitektur *client-server*, tanpa memerlukan aplikasi tambahan.



Gambar 3. Gambar arsitektur sistem

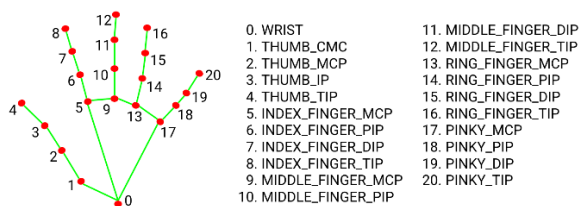
Gambar 3 menggambarkan arsitektur sistem kendali gestur berbasis *web*. Kamera laptop berfungsi sebagai sensor untuk menangkap citra tangan pengguna. Data citra tersebut diproses oleh aplikasi *web* yang dijalankan di *browser*, di mana MediaPipe melakukan ekstraksi *landmark* tangan secara *real-time*. Hasil deteksi *landmark* kemudian diklasifikasikan menjadi gestur ON atau OFF dengan menggunakan perhitungan rasio posisi antar titik. Apabila gestur terdeteksi valid, *browser* mengirimkan perintah berupa HTTP request ke ESP32 yang berperan sebagai server. ESP32 menerima instruksi tersebut dan mengaktifkan atau mematikan lampu LED sesuai perintah yang diterima. Diagram tersebut menunjukkan alur logis sistem secara keseluruhan dan hubungan antar komponen utama dalam arsitektur *client-server*.



Gambar 4. Alur kerja sistem

Alur kerja sistem dimulai dari akuisisi citra tangan, pemrosesan *landmark* oleh MediaPipe, klasifikasi gestur menggunakan fungsi perbandingan rasio jarak tangan, hingga pengiriman perintah ke ESP32 untuk menyalakan atau mematikan lampu LED.

Sistem bekerja dalam pola komunikasi *client-server*, di mana aplikasi *web* berfungsi sebagai pendeteksi gestur dan pengirim perintah, sedangkan mikrokontroler ESP32 menerima dan mengeksekusi instruksi pada lampu LED. Proses dimulai dari kamera yang menangkap citra tangan, kemudian MediaPipe melakukan deteksi 21 titik *landmark* dan menghasilkan klasifikasi gestur.



Gambar 5. Pemetaan 21 titik landmark [1]

Pada proses klasifikasi gestur tangan, sistem memanfaatkan koordinat tiga dimensi dari 21 titik *landmark* yang disediakan oleh MediaPipe seperti yang ada pada Gambar 5. Setiap titik dinyatakan dalam bentuk koordinat (x, y, z) , merepresentasikan posisi ruas jari dan pergelangan dalam koordinat tiga dimensi, sehingga hubungan antar-titik dapat dihitung secara geometris. Untuk menentukan apakah suatu jari berada dalam keadaan terbuka atau tertutup, sistem menghitung jarak antara dua titik *landmark* menggunakan rumus *Euclidean Distance* sebagai berikut:

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (1)$$

di mana $d(P_1, P_2)$ merupakan jarak antara dua titik *landmark*, sedangkan (x_1, y_1, z_1) dan (x_2, y_2, z_2) masing-masing menyatakan koordinat titik pertama dan titik kedua.

Selanjutnya, jarak tersebut digunakan untuk memperoleh rasio panjang relatif jari terhadap posisi pergelangan tangan. Rasio dihitung dengan membandingkan jarak antara ujung jari dengan pergelangan tangan terhadap jarak pangkal jari dengan pergelangan tangan. Rasio ini dinyatakan sebagai berikut:

$$R_j = \frac{d(\text{ujung_jari, pergelangan})}{d(\text{pangkal_jari, pergelangan})} \quad (2)$$

Jari dikategorikan terbuka apabila nilai rasio $R_j > 1,5$. Gestur *ON* diidentifikasi ketika seluruh lima jari memenuhi kondisi ini, sedangkan apabila satu atau lebih jari memiliki nilai $R_j \leq 1,5$, sistem akan mengklasifikasikannya sebagai gestur *OFF*. Pendekatan berbasis rasio ini bersifat invarian terhadap ukuran tangan, memiliki kompleksitas komputasi yang rendah, serta mendukung kinerja *real-time* pada antarmuka *web*, sehingga sesuai untuk

aplikasi kendali perangkat IoT. Hasil keluaran klasifikasi berupa perintah *ON* atau *OFF* dikirimkan ke ESP32 melalui protokol HTTP.

3.3. Metode Pengujian Sistem

Metode pengujian pada penelitian ini dilakukan melalui dua pendekatan utama, yaitu pengujian fungsional dan pengujian akurasi, yang masing-masing bertujuan untuk memastikan bahwa sistem bekerja sesuai rancangan serta mampu mengenali gestur tangan.

Pengujian fungsional dilakukan terlebih dahulu untuk memverifikasi bahwa seluruh rangkaian alur sistem berjalan tanpa kesalahan. Tahapan yang diuji meliputi proses deteksi *landmark* tangan oleh MediaPipe, klasifikasi gestur berdasarkan perhitungan rasio panjang jari terhadap pergelangan tangan, pengiriman instruksi *ON* atau *OFF* melalui protokol HTTP dari aplikasi *web*, serta eksekusi aktuasi pada LED oleh mikrokontroler ESP32. Sistem dinyatakan berfungsi dengan baik apabila seluruh proses tersebut dapat berjalan tanpa *error* dan LED merespons perintah secara *real-time* sesuai hasil klasifikasi.

Setelah pengujian fungsional terpenuhi, dilakukan pengujian akurasi untuk mengevaluasi kemampuan sistem dalam mengenali gestur *ON* dan *OFF* pada kondisi jarak serta tingkat pencahayaan tertentu. Pengujian dilakukan pada rentang jarak 30–290 cm dengan interval 10 cm dengan area pengujian berupa lintasan tegak lurus di depan kamera dengan posisi salah satu telapak tangan menghadap ke kamera, serta melibatkan dua sumber pencahayaan yaitu lampu 10 Watt dan 3 Watt. Pada setiap titik jarak, dilakukan sebanyak 30 percobaan untuk gestur *ON* dan 30 percobaan untuk gestur *OFF*, sehingga setiap kombinasi kondisi menghasilkan total 60 percobaan.

Dalam penelitian ini digunakan kriteria keberhasilan bahwa suatu gestur dianggap berhasil dikenali apabila minimal 25 dari 30 percobaan ($\geq 80\%$) terdeteksi dengan benar. Ambang batas ini dipilih karena secara umum digunakan dalam evaluasi sistem berbasis visi komputer yang mengharuskan stabilitas deteksi dalam kondisi tertentu. Nilai akurasi diperoleh menggunakan rumus berikut:

$$\text{Akurasi}(\%) = \frac{\text{Jumlah Deteksi Benar}}{\text{Jumlah Percobaan}} \times 100\% \quad (3)$$

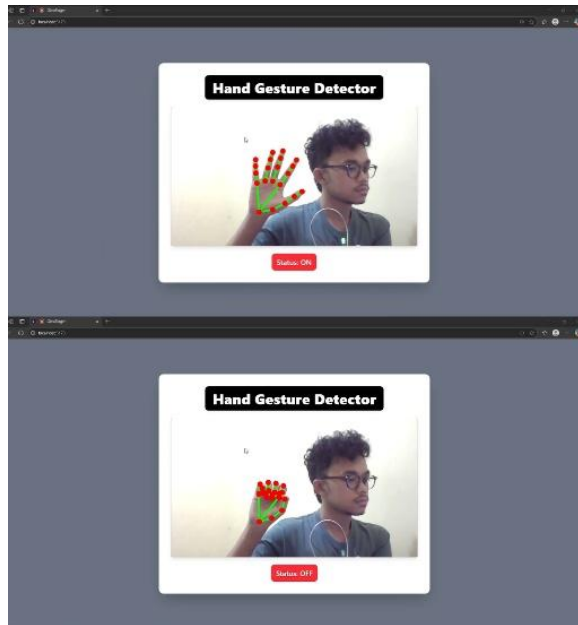
Rumus tersebut menghitung persentase keberhasilan sistem dalam mengenali gestur pada setiap kondisi pengujian. Hasil pengujian akurasi kemudian dijadikan dasar untuk menilai stabilitas deteksi gestur pada variasi jarak dan pencahayaan.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Sistem

Realisasi sistem dilakukan melalui orkestrasi antara antarmuka berbasis *web* dengan mikrokontroler ESP32 yang bertindak sebagai unit kendali aktuator. Pada sisi perangkat lunak, sistem mendayagunakan pustaka MediaPipe Hand untuk melakukan ekstraksi 21 titik *landmark* tangan secara *real-time*. Mekanisme

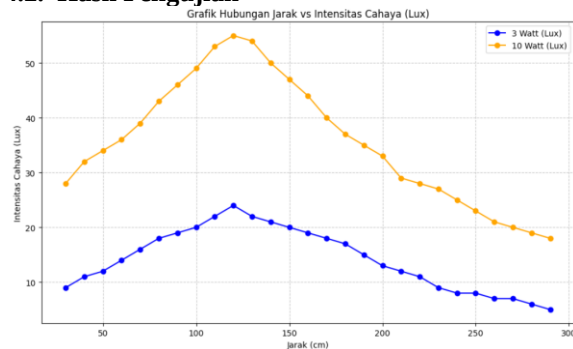
klasifikasi gestur beroperasi berdasarkan komparasi proporsi geometris jari, yang membedakan kondisi 'ON' (seluruh jari terbuka) dan 'OFF' (jari tertutup).



Gambar 6. Deteksi gestur

Sebagaimana ditampilkan pada Gambar 6, data instruksi dikirimkan menggunakan protokol HTTP POST menuju ESP32 untuk memicu perubahan status GPIO. Hal ini memastikan responsivitas sistem dalam memvisualisasikan interaksi fisik tangan menjadi perubahan status nyala lampu secara sinkron.

4.2. Hasil Pengujian



Gambar 7. Grafik hubungan jarak dan intensitas cahaya

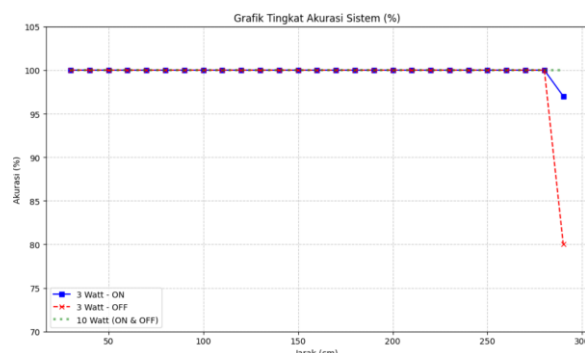
Merujuk pada visualisasi data di Gambar 7, terlihat pola distribusi intensitas cahaya terhadap jarak tangan dari kamera. Pada kedua skenario lampu, intensitas cahaya menunjukkan tren kenaikan seiring bertambahnya jarak tangan dari kamera, mencapai titik puncak pada jarak 120 cm, kemudian mengalami penurunan secara gradual hingga jarak terjauh 290 cm. Pada lampu 10 Watt, intensitas maksimum tercatat sebesar 55 lux di titik 120 cm dan nilai minimum 18 lux pada jarak terjauh 290 cm. Sementara itu, sumber cahaya 3 Watt memiliki intensitas puncak yang lebih

rendah di angka 24 lux, dengan nilai minimum yang sangat rendah, yakni 5 lux pada titik akhir pengujian

Evaluasi selanjutnya meninjau performa akurasi sistem dalam mendeteksi gestur 'ON' dan 'OFF' berdasarkan variabel jarak dan pencahayaan tersebut. Rincian data kuantitatif dan tren stabilitas sistem disajikan dalam tabel dan grafik sebagai berikut:

Tabel 1. Data hasil uji

Jarak (cm)	3 watt	Kondisi		10 watt	Kondisi	
	Lux (lx)	On (%)	Off (%)	Lux (lx)	On (%)	Off (%)
30	9	100	100	28	100	100
40	11	100	100	32	100	100
50	12	100	100	34	100	100
60	14	100	100	36	100	100
70	16	100	100	39	100	100
80	18	100	100	43	100	100
90	19	100	100	46	100	100
100	20	100	100	49	100	100
110	22	100	100	53	100	100
120	24	100	100	55	100	100
130	22	100	100	54	100	100
140	21	100	100	50	100	100
150	20	100	100	47	100	100
160	19	100	100	44	100	100
170	18	100	100	40	100	100
180	17	100	100	37	100	100
190	15	100	100	35	100	100
200	13	100	100	33	100	100
210	12	100	100	29	100	100
220	11	100	100	28	100	100
230	9	100	100	27	100	100
240	8	100	100	25	100	100
250	8	100	100	23	100	100
260	7	100	100	21	100	100
270	7	100	100	20	100	100
280	6	100	100	19	100	100
290	5	97	80	18	100	100



Gambar 8. Grafik hubungan jarak dan akurasi sistem

Berdasarkan Tabel 1 dan Gambar 8, terlihat perbedaan performa akurasi sistem pada kedua kondisi pencahayaan. Pada skenario pencahayaan 10 Watt, sistem menunjukkan stabilitas tinggi dengan mempertahankan akurasi sempurna 100% pada seluruh rentang jarak uji 30–290 cm. Sebaliknya, pada skenario 3 Watt, konsistensi akurasi 100% hanya

bertahan hingga jarak 280 cm. Pada jarak ekstrem 290 cm, terjadi penurunan performa di mana akurasi gestur 'ON' turun menjadi 97%, dan gestur 'OFF' mengalami penurunan yang signifikan hingga level 80%.

4.3. Pembahasan

Berdasarkan hasil pengujian, bentuk grafik intensitas cahaya yang naik hingga jarak 120 cm dan turun setelahnya disebabkan oleh tata letak pengujian, di mana posisi sumber cahaya berada tegak lurus tepat di jarak 120 cm dari kamera. Penurunan intensitas cahaya yang tajam saat objek bergerak menjauh dari lampu menuju 290 cm adalah konsekuensi alami dari perambatan cahaya, di mana intensitas cahaya akan melemah secara signifikan seiring bertambahnya jarak dari sumber cahaya. Dalam konteks sistem ini, penurunan drastis tersebut menjadi tantangan utama bagi sensor visual.

Disparitas performa yang terjadi pada jarak 290 cm antara skenario 10 Watt dan 3 Watt menunjukkan adanya ambang batas sensitivitas pada algoritma deteksi. Pada skenario 10 Watt, meskipun intensitas cahaya turun akibat jarak, nilai minimum yang diterima sensor kamera masih berada di angka 18 lux. Tingkat iluminasi ini terbukti masih memadai bagi sensor kamera untuk mempertahankan kontras visual yang baik, sehingga algoritma MediaPipe dapat membedakan *landmark* antar ruas jari secara presisi.

Sebaliknya, pada skenario 3 Watt di jarak 290 cm, intensitas cahaya jatuh hingga ke level 5 lux. Pada kondisi minim cahaya ini, sensor kamera mengalami penurunan kualitas citra akibat munculnya *noise* yang signifikan. *Noise* ini mengaburkan detail tepi jari. Akibatnya, sistem kesulitan membedakan antara jari yang menekuk sedikit dengan jari lurus, yang menyebabkan akurasi deteksi gestur 'OFF' turun drastis hingga 80%. Kegagalan yang lebih tinggi pada gestur 'OFF' dibandingkan gestur 'ON' mengindikasikan bahwa kondisi jari tertutup membutuhkan detail bayangan yang lebih tegas untuk dapat dikenali oleh model.

Temuan ini menegaskan bahwa dalam implementasi sistem kendali berbasis visi komputer, variabel pencahayaan memegang peranan yang lebih dominan dibandingkan variabel jarak semata. Jika intensitas cahaya lingkungan turun di bawah ambang batas 10 lux, sistem memerlukan mekanisme kompensasi pencahayaan agar akurasi tetap terjaga.

5. KESIMPULAN DAN SARAN

Penelitian ini telah sukses merealisasikan integrasi sistem kendali nirkabel berbasis gestur tangan yang memadukan keandalan *framework* MediaPipe dengan mikrokontroler ESP32, menghasilkan responsivitas pengendalian perangkat yang optimal. Temuan empiris menunjukkan bahwa sistem mampu mempertahankan akurasi absolut pada kondisi pencahayaan yang memadai, namun teridentifikasi adanya degradasi performa signifikan

pada jarak 290 cm saat intensitas cahaya menyentuh angka 5 lux. Secara akademis, hasil ini memberikan kontribusi penting dalam memetakan ambang batas kritis iluminasi yang terbukti menjadi determinan lebih dominan dibandingkan variabel jarak terhadap stabilitas sistem interaksi berbasis visi komputer. Guna mengatasi batasan tersebut, pengembangan selanjutnya sangat disarankan untuk menerapkan algoritma klasifikasi berbasis *Machine Learning* agar sistem mampu mengenali variasi gestur yang lebih kompleks secara dinamis, serta mengintegrasikan modul sensor tambahan untuk mengkompensasi penurunan akurasi pada lingkungan dengan intensitas cahaya rendah.

DAFTAR PUSTAKA

- [1] Indriani, M. Harris, and A. S. Agoes, "Applying Hand Gesture Recognition for User Guide Application Using MediaPipe," in *Advances in Engineering Research (Proceedings of the 2nd International Seminar of Science and Applied Technology - ISSAT 2021)*, vol. 207, pp. 101–108, 2021.
- [2] G. Sung *et al.*, "On-device Real-time Hand Gesture Recognition," *arXiv preprint arXiv:2111.00038*, 2021.
- [3] V. Adithya and R. Rajesh, "A Deep Convolutional Neural Network Approach for Static Hand Gestur Recognition," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 2353–2361. doi: 10.1016/j.procs.2020.04.255.
- [4] C. Ferry, M. Saputra, and W. Sulisty, "Alat Keamanan Depan Rumah Berbasis Internet of Things (IoT) Menggunakan ESP32-CAM yang Terintegrasi dengan Face Detection dan Telegram," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 8, no. 1, p. 2024, 2024, doi: 10.35870/jti.
- [5] R. Samsinar *et al.*, "Sistem Pendeteksi Kurir Menggunakan Smart Closed Circuit Television (CCTV) Berbasis Internet Of Things (IoT) dengan Media Komunikasi Bot Telegram (Studi Kasus : Rumah Indekost)," *RESISTOR (Elektronika Kendali Telekomunikasi Tenaga Listrik Komputer)*, vol. 6, no. 1, pp. 47–54.
- [6] M. Nizam, H. Yuana, and Z. Wulansari, "Mikrokontroler ESP 32 Sebagai Alat Monitoring Pintu Berbasis Web," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 2, pp. 767–772, 2022.
- [7] A. Ramschie *et al.*, "Pemanfaatan ESP32 Pada Sistem Keamanan Rumah Tinggal Berbasis IoT," in *Prosiding The 12th Industrial Research Workshop and National Seminar*, Bandung, 2021.
- [8] C. Y. Yang *et al.*, "Smart Control of Home Appliances Using Hand Gestur Recognition in an IoT-Enabled System," *Applied Artificial Intelligence*, vol. 37, no. 1, 2023, doi: 10.1080/08839514.2023.2176607.

- [9] B. Artono and R. G. Putra, "Penerapan Internet Of Things (IoT) Untuk Kontrol Lampu Menggunakan Arduino Berbasis Web," *Jurnal Teknologi Informasi dan Terapan*, vol. 5, no. 1, pp. 9–16, Apr. 2019, doi: 10.25047/jtit.v5i1.73.
- [10] F. Zhang *et al.*, "MediaPipe Hands: On-device Real-time Hand Tracking." [Online]. Available: <https://mediapipe.dev>.
- [11] M. A. Fakhruddin, H. Pratikno, Musayyanah, and W. I. Kusumawati, "Kontrol Level Kecepatan Kipas Melalui Deteksi Gestur Jari Tangan Menggunakan MediaPipe dan Faster-RCNN," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 6, pp. 1295–1302, Dec. 2023, doi: 10.25126/jtiik.2023107345.
- [12] A. D. Agustiani, M. R. Sholahuddin, S. M. Putri, and P. Hidayatullah, "Penggunaan MediaPipe untuk Pengenalan Gestur Tangan Real-Time dalam Pengendalian Presentasi," *Media Jurnal Informatika*, vol. 16, no. 2, p. 147, Dec. 2024, doi: 10.35194/mji.v16i2.4788.
- [13] M. R. P. Nautica, "Hand Gesture Detection Sebagai Alat Bantu Ajar Berhitung Menggunakan MediaPipe dan Convolutional Neural Network Secara Realtime," Tugas Akhir, Universitas Dinamika, Surabaya, 2022.
- [14] D. Khairianto and R. Firdaus, "Penerapan Hand Gesture Recognition Sebagai Media Kontrol Presentasi Aplikasi PowerPoint," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 2, 2024.
- [15] A. M. Chalik, A. Qowy, F. Hanafi, and A. Nuraminah, "Mouse Tracking Tangan dengan Klasifikasi Gestur Menggunakan OpenCV dan MediaPipe," *JUITIK*, vol. 1, no. 2, 2021, [Online]. Available: <http://journal.sinov.id/index.php/juitik/indexHalamanUTAMAJurnal>; <https://journal.sinov.id/index.php>