

KONTROL *LEVEL* KECEPATAN KIPAS MELALUI DETEKSI GESTUR JARI TANGAN MENGGUNAKAN MEDIAPIPE DAN FASTER-RCNN

Muhammad Aldi Fakhruddin¹, Heri Pratikno^{*2}, Musayyanah³, Weny Indah Kusumawati⁴

^{1,2,3,4}Universitas Dinamika, Surabaya

Email: ¹aldifakhruddin0@gmail.com, ²heri@dinamika.ac.id, ³musayyanah@dinamika.ac.id,

⁴weny@dinamika.ac.id

^{*}Penulis Korespondensi

(Naskah masuk: 05 Juni 2023, diterima untuk diterbitkan: 27 November 2023)

Abstrak

Interaksi antara manusia dan komputer saat ini lebih interaktif, responsif dan intuitif, di masa lalu proses interaksi tersebut diperlukan kontak secara fisik atau menggunakan sensor-sensor elektronik. Pada penelitian ini interaksi antara manusia dan komputer atau peralatan elektronik tidak diperlukan kontak fisik maupun melalui sensor karena dilakukan secara *computer vision* hanya menggunakan *webcam* sehingga proses interaksinya lebih natural. Penerapan mikrokontroler sebagai *backbone* utama teknologi berbasis *Internet of Things* di era *Industry 4.0*, bertujuan untuk mempermudah pekerjaan manusia terutama dukungan layanan di dunia industri. Pada era *Society 5.0* semua penerapan teknologi yang ada tujuan utamanya tidak hanya mempermudah pekerjaan manusia tetapi bagaimana teknologi tersebut bisa lebih mengerti dan memahami manusianya maka disitulah diterapkan *Artificial Intelligence*. Dalam penelitian ini diterapkan sistem kontrol interaksi antara pengguna dan komputer untuk pengaturan *level* kecepatan putaran kipas angin secara otomatis dan *realtime* berbasis teknologi *computer vision for deep learning* melalui deteksi bentuk gestur jari tangan kanan dan gestur jari tangan kiri menggunakan *webcam*. Mikrokontroler yang digunakan pada penelitian ini adalah Arduino Uno, sedangkan penerapan *computer vision for deep learning* menggunakan *framework* MediaPipe dan Faster-RCNN. MediaPipe berfungsi untuk mendeteksi bentuk gestur fitur jari kedua tangan dan Faster-RCNN digunakan untuk proses klasifikasi empat bentuk gestur jari tangan untuk mematikan kipas angin atau menghidupkan kipas angin dengan kecepatan putarannya pada *level* 1, *level* 2 atau *level* 3. Hasil pengujian akurasi rata-rata deteksi gestur jari tangan menggunakan MediaPipe pada jarak 10 cm (41,6%), jarak 50 cm (85,35%), jarak 100 cm (71,68%), dan jarak 175 cm (69,33%). Sedangkan hasil pengujian Faster-RCNN mempunyai akurasi klasifikasi rata-rata pada jarak 10 cm (36%), jarak 50 cm (30,75%), jarak 100 cm (18,68 %), dan jarak 175 cm (14,83%).

Kata kunci: HCI, Computer Vision, Deep Learning, MediaPipe, Faster-RCNN, Arduino Uno

FAN SPEED LEVEL CONTROL THROUGH FINGER GESTURE DETECTION USING MEDIAPIPE AND FASTER-RCNN

Abstract

Interaction between humans and computers is now more interactive, responsive and intuitive, in the past the interaction process required physical contact or using electronic sensors. In this study, the interaction between humans and computers or electronic equipment does not require physical contact or through sensors because it is done in *computer vision* using only a *webcam* so that the interaction process is more natural. The application of microcontrollers as the main backbone of *Internet of Things*-based technology in the *Industry 4.0* era, aims to facilitate human work, especially service support in the industrial world. In the era of *Society 5.0*, all applications of technology that have the main goal are not only to facilitate human work but how technology can better understand and understand humans, so that's where *Artificial Intelligence* is applied. In this study, an interaction control system was applied between the user and the computer to adjust the fan speed level automatically and in real time based on *computer vision* technology for deep learning through the detection of the shape of the right hand finger gesture and the left hand finger gesture using a *webcam*. The microcontroller used in this study is Arduino Uno, while the application of *computer vision* for deep learning uses the MediaPipe and Faster-RCNN frameworks. MediaPipe serves to detect the shape of the finger feature gestures of both hands and Faster-RCNN is used to process the classification of four finger gestures to turn off the fan or turn on the fan with its rotational speed at level 1, level 2 or level 3. The results of the average accuracy test detection of finger gestures using the MediaPipe at a distance of 10 cm (37%), a distance of 50 cm (70%), a distance of 100 cm (54.7%), and a distance

of 175 cm (63.3%). While the Faster-RCNN test results have an average classification accuracy at a distance of 10 cm (34%), a distance of 50 cm (24%), a distance of 100 cm (8.7%), and a distance of 175 cm (4.7%).

Keywords: HCI, Computer Vision, Deep Learning, MediaPipe, Faster-RCNN, Arduino Uno

1. PENDAHULUAN

Perkembangan yang pesat dalam bidang *computer vision* (visi komputer) telah berhasil menjembatani kesenjangan interaksi antara manusia dan komputer, mesin atau peralatan elektronik dalam beberapa tahun terakhir karena lebih efektif dan alami. Proses konvergensi visi komputer untuk interaksi manusia-komputer dapat berjalan secara *realtime* sehingga dapat dimanfaatkan untuk proses pengenalan gerakan, pengenalan objek, analisis gerakan, dan banyak lagi.

Tahapan proses komputasi dalam visi komputer meliputi akuisisi data, pemrosesan, pemeriksaan, pemahaman citra digital, dan ekstraksi data berkualitas lebih tinggi. Awalnya, teknik *computer vision* digunakan untuk mengidentifikasi dan mengenali objek yang ada dalam video dan gambar, tetapi saat ini digunakan secara luas di bidang kontrol transportasi, lingkungan, medis, keamanan dan sebagainya.

Secara umum sinyal biometrik (elektromiografi, EMG) dapat ditangkap secara non-invasif melalui penempatan elektroda di tangan manusia, pada penelitian (Ramadoss *et al.*, 2021) mengusulkan prediksi pengenalan gerakan manusia secara *computer vision* melalui penerapan interaksi manusia dan komputer noninvasif yang dioptimalkan (ONIHCI) dan metode *Average Intrinsic Mode Function* (AIMF). Metode AIMF digunakan untuk mengurangi faktor *noise* pada sinyal EMG sebesar 7,2% dengan peningkatan akurasi sebesar 97,2%.

Penerapan awal mula bidang *computer vision* adalah proses untuk mengenali wajah, ekspresi wajah, mendeteksi jenis kelamin melalui wajah, memprediksi usia seseorang melalui wajah kemudian berkembang dalam proses pengenalan *pose* bentuk tubuh, gerakan gestur dari anggota badan serta digunakan untuk deteksi detak jantung atau denyut nadi. Salah satu contoh penelitian (Arsal, Wardijono and Anggraini, 2020) menerapkan *computer vision* berbasis pengenalan wajah untuk sistem keamanan pintu akses pegawai bank menggunakan *deep learning* dengan algoritma *Convolutional Neural Network* (CNN) dengan total dataset sebanyak 350 data wajah menghasilkan persentase akurasi 95%.

Penelitian lain (Asmaleni, Hamdani and Sakti, 2020) yang linier dengan penelitian yang dilakukan oleh penulis dengan topik sistem kontrol kipas angin dan lampu otomatis berbasis saklar suara menggunakan Arduino Uno melalui *smartphone* Android dengan koneksi melalui *Bluetooth* HC-05 dengan persentase keberhasilannya 82,5%.

Implementasi *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). Hasil dari penelitian tersebut metode MediaPipe mempunyai akurasi 90,99% dalam mendeteksi bentuk gestur kesepuluh jari tangan, sedangkan CNN mempunyai akurasi klasifikasi sebesar 38,67%.

Pada penelitian ini mengusulkan sebuah penerapan *framework* MediaPipe dan Faster-RCNN yang digunakan untuk proses interaksi antara pengguna, komputer, Arduino Uno dan kipas angin secara otomatis melalui *webcam*. Proses interaksi tersebut berjalan dalam waktu nyata (*realtime*) berbasis *computer vision for deep learning*. Dikarenakan jumlah *level* kecepatan putaran kipas hanya ada empat kondisi tombol (On/off, kecepatan *level 1*, *level 2* dan *level 3*) maka juga hanya ada empat bentuk gestur jari tangan yang digunakan.

Terdapat empat hal utama yang akan di analisis pada penelitian ini sebagai parameter uji, yaitu: persentase akurasi terhadap perubahan jarak (0-300 cm) antara *webcam* dengan pengguna, akurasi deteksi empat gestur jari tangan, akurasi deteksi empat gestur jari tangan terhadap perubahan jarak serta besarnya nilai *Frame per Second* (FPS). Visi komputer memiliki potensi membawa banyak informasi dengan cara yang efektif dengan biaya rendah, sehingga mempunyai modalitas penginderaan yang sangat menarik untuk dikembangkan pada skema cerdas interaksi manusia-komputer.

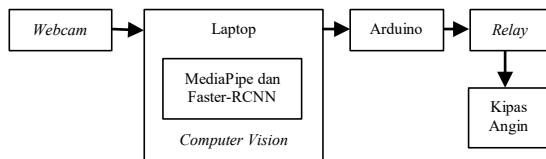
2. METODE PENELITIAN

2.1. Perancangan Perangkat Keras

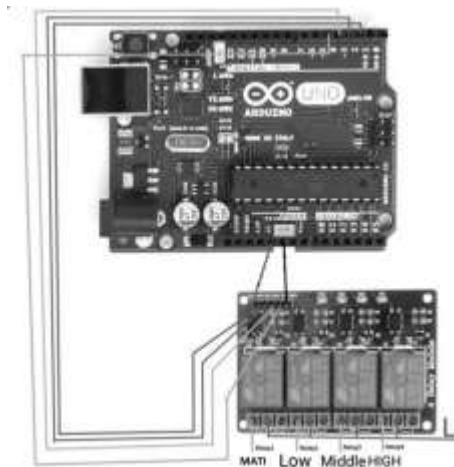
Tahapan kerja yang dilakukan pada penelitian ini akan dibahas pada metode penelitian sebagaimana terlihat pada Gambar 1 yang menunjukkan desain perancangan perangkat keras. Langkah pertama, mendeteksi gestur jari tangan kanan atau jari tangan kiri oleh *webcam*, langkah kedua hasil deteksi *webcam* tersebut kemudian diproses oleh *laptop* menggunakan MediaPipe untuk dilakukan pengenalan bentuk fitur jari tangan dan metode Faster-RCNN untuk proses klasifikasi bentuk gestur jari tangan yang hasilnya ditampilkan di layar *laptop*.

Berikutnya langkah ketiga data hasil komputasi pada langkah kedua dikirim ke Arduino Uno menggunakan komunikasi antarmuka serial, pada langkah keempat data dari Arduino mengaktifkan (ON, *Normally Open*) atau menon-aktifkan (OFF, *Normally Close*) *relay*. Selanjutnya pada langkah kelima aktivasi terhadap 4 buah *relay* akan mengeksekusi tombol *level* kecepatan putaran kipas

angin dengan bentuk gambar skematik koneksinya seperti pada Gambar 2.



Gambar 1. Blok diagram perancangan perangkat keras



Gambar 2. Skematik diagram koneksi Arduino Uno dan relay

2.2. Instalasi *Environment*

Diperlukan proses instalasi dan konfigurasi *environment* agar kebutuhan komputasi perangkat lunak bisa berjalan dengan baik karena semua *library* yang diperlukan sudah terinstal. Adapun program dan *library* yang diperlukan pada penelitian ini Python 3.7.0, *plugin* dan *library* pada terminal *pip* dan Anaconda, diantaranya: *protobuf*, *pillow*, *lxml*, *Cython*, *contextlib2*, *jupyter*, *matplotlib*, *pandas*, *opencv-python*, *MediaPipe*, *cvzone*, *pyserial*. Terdapat *string terminalRead* pada mikrokontroler Arduino untuk pembacaan data dari terminal serial Arduino dalam bentuk data *string*.

2.3. Dataset

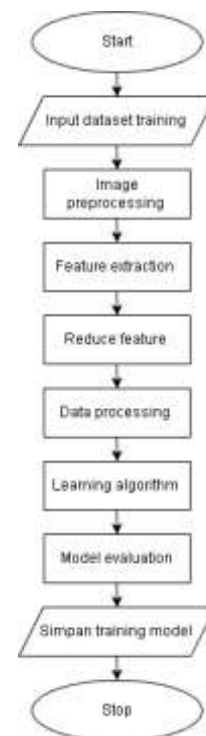
Dataset yang digunakan pada penelitian ini sebanyak 4.800 citra dari 10 gestur jari tangan kanan dan tangan kiri yang terdiri dari citra asli terlihat pada Gambar 3 dan citra digital dari hasil proses augmentasi. Augmentasi merupakan sebuah metode untuk memperbanyak jumlah *dataset* melalui proses komputasi citra gestur jari asli menjadi citra digital hasil dari proses pengolahan citra dengan cara ditransformasi dalam beberapa bentuk tetapi tidak merubah makna atau arti dari citra tersebut. Proses transformasi citra dari augmentasi bisa dalam bentuk *rotate*, *size*, *zoom*, *flip*, *mirror* dan sebagainya.

Tujuan utama memperbanyak atau menambah jumlah *dataset training* maupun validasi adalah untuk meningkatkan nilai akurasi pada saat tahapan proses *testing* atau klasifikasi. dari Faster-RCNN. Perincian

dari *dataset* sejumlah 4.800, untuk proses *training* sebanyak 3.840 citra atau 80% dari jumlah citra *dataset* dan untuk validasi 960 citra atau setara dengan 20% dari jumlah citra *dataset*. Jumlah *dataset* untuk proses *training* harus lebih besar dari jumlah data untuk validasi, agar sistem dapat “belajar” (Azis *et al.*, 2020).



Gambar 3. Dataset jari tangan

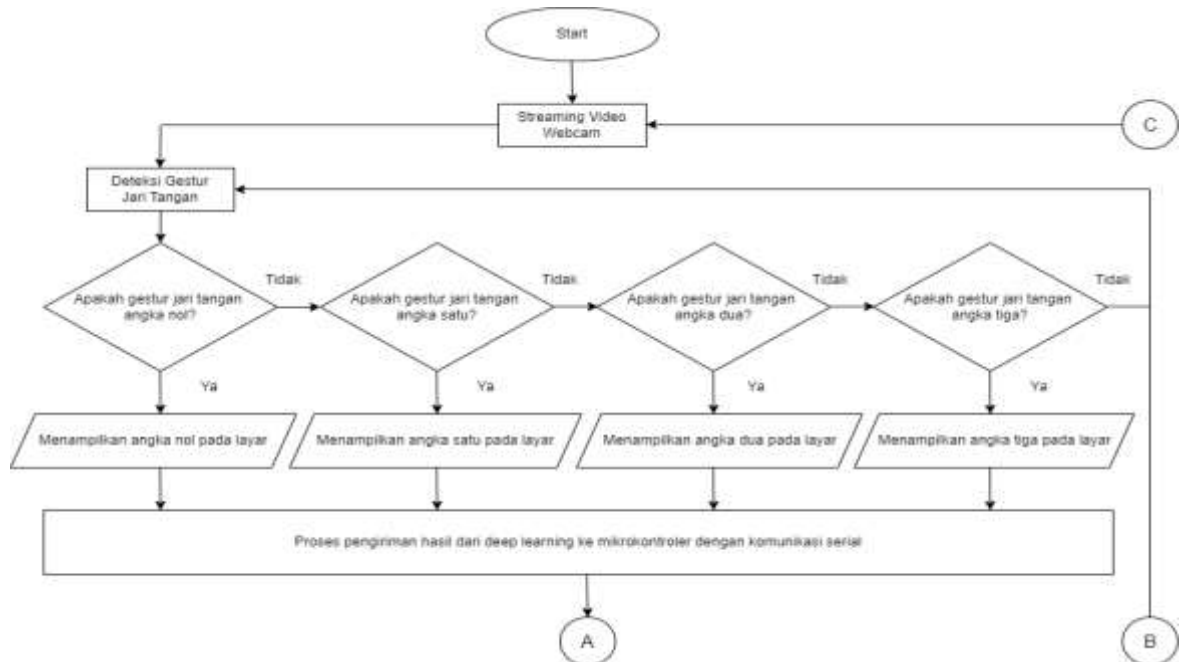


Gambar 4. Flowchart proses training dari dataset

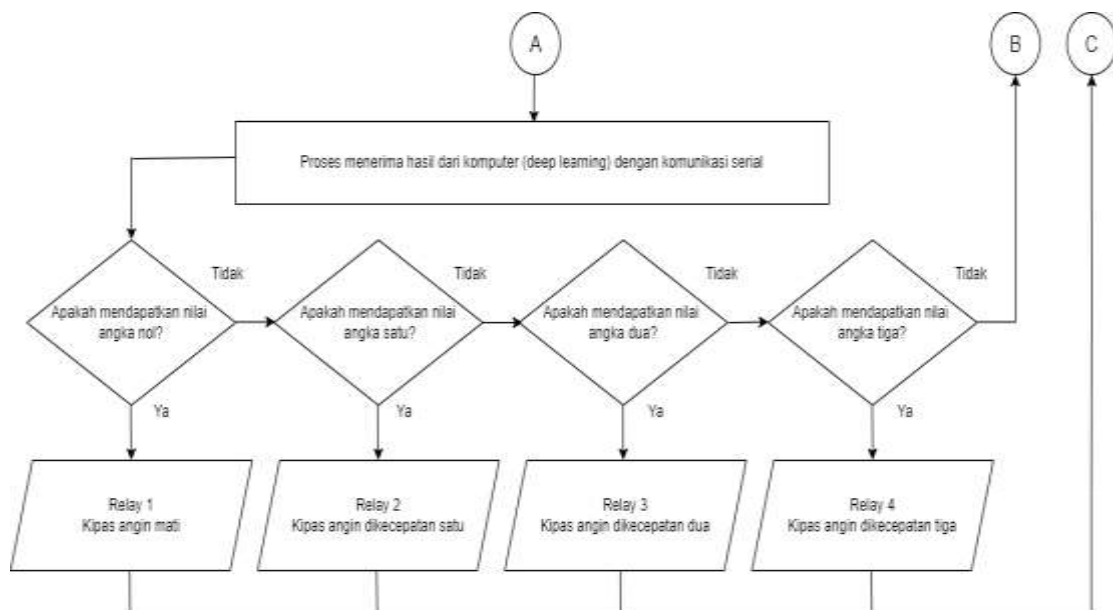
Tahapan proses training pada Faster-RCNN dapat dilihat pada *flowchart* sebagaimana tampak pada Gambar 4, proses *training* ini dilakukan sebanyak 8.000 *step* dan hasil dari proses evaluasi model *training* disimpan dalam model dengan format file ber-ekstensi *.CKPT*. *Feature extraction* untuk mengenali ciri khusus suatu obyek, sedangkan *reduce feature* untuk memilih fitur yang berpengaruh serta mengesampingkan fitur yang tidak diinginkan.

2.4. Proses Testing Program Deep Learning

Tahapan proses testing program pada lingkungan *deep learning* dapat dilihat pada *flowchart* Gambar 5, dimana pada tahapan pertama *webcam* akan mendeteksi dan melacak keberadaan jari tangan kanan maupun jari tangan kiri.



Gambar 5. Flowchart proses di Deep Learning



Gambar 6. Flowchart proses di Arduino Uno

Apabila terdeteksi jari tangan maka sistem akan mengenali bentuk gestur dari jari tangan tersebut, pada saat terdeteksi jari tangan menggenggam maka pada saat terdeteksi jari tangan menggenggam maka pada layar *laptop* akan ditampilkan angka 0, maka angka 0 tersebut dikirim ke Arduino Uno melalui kabel serial kemudian data tersebut diproses oleh Arduino Uno serta diterjemahkan dalam bentuk sebuah aksi untuk mematikan kipas. Jika yang terdeteksi bentuk gestur jari tangan dengan jumlah 1 jari, 2 jari atau 3 jari tangan maka data tersebut akan diartikan untuk menjalankan kecepatan putaran baling-baling kipas angin pada *level 1*, *level 2* atau *level* kecepatan ke-3, proses tersebut diulang secara terus-menerus (*looping*) sampai program ditutup.

2.6. Flowchart Program Mikrokontroler

Tampilan gambar *flowchart* proses yang ada di Arduino Uno tampak pada Gambar 6 di bawah, data yang dikirim hasil komputasi *computer vision for deep learning* dari *laptop* dikirim ke Arduino Uno dalam bentuk data *string*, yang mana format data *string* tersebut memerlukan *library pyserial* melalui jenis kabel USB Serial. Data *string* tersebut oleh Arduino Uno diperbandingkan dengan sintak *if else* yang akan mendapatkan nilai '0', '1', '2', atau '3' berdasarkan data tersebut akan mengakses aktuator berupa *relay*, *relay-relay* tersebut fungsinya sebagai pengganti tombol-tombol manual kecepatan putaran yang ada di kipas. Proses tersebut diatas akan

diulang-ulang secara terus menerus (*looping*) hingga program ditutup.

Deklarasi variabel *terminalRead* untuk menyimpan nilai atau karakter yang berasal dari komunikasi serial dengan *laptop*, variabel *serialComm* berfungsi mengetahui koneksi *port* USB Serial dan *baud-rate* dari Arduino Uno, variabel *hand_mode* untuk mendeteksi kode tangan, nilai string "1" adalah tangan kanan serta variabel *counter* untuk menghitung jumlah jari tangan.

2.7. MediaPipe

MediaPipe adalah *framework open source* yang dikembangkan oleh Google pada tahun 2019 untuk membangun *pipelines* pengolahan data persepsi dari berbagai macam bentuk format data, diantaranya: audio, video, *pose* dan NLP. Pada penelitian ini menerapkan *framework* MediaPipe *hand* untuk mendeteksi dan menjumlahkan 21 *keypoints* pada ruas jari tangan tampak pada Gambar 7.

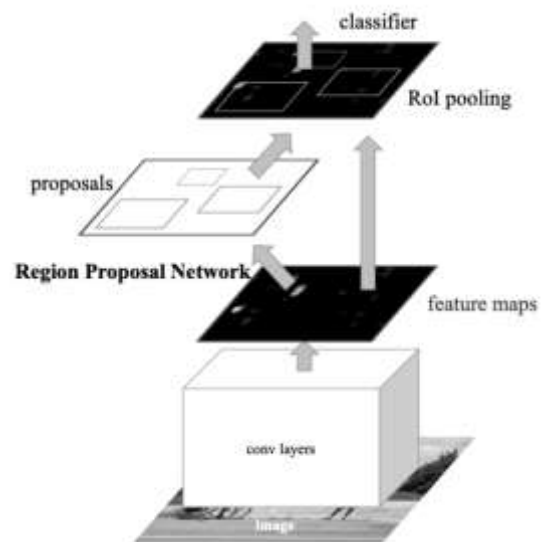


Gambar 7. Bobot *keypoints* pada ruas jari tangan

Penelitian (Aliyah, 2022) menggunakan MediaPipe dan LSTM untuk menerjemahkan gerakan bahasa isyarat menjadi kosa kata dengan jumlah *dataset* 90 menghasilkan akurasi 77,8%, *recall*, *precision* dan *F1-Score* 0,9. Peneliti lain (Kurnawan, 2022) untuk membandingkan alat sensor goniometer manual dengan hasil deteksi dari MediaPipe dengan jarak 100 cm mendapatkan selisih sudut 3,94% dan nilai error 8%. Sedangkan penelitian *gesture recognition* (Makahaube, Sambul and Sompie, 2021) menggunakan MediaPipe yang dimanfaatkan mengontrol *mouse* secara virtual dari *webcam* hanya dapat mendeteksi 1 objek tangan saja.

2.8. Faster-RCNN

Faster R-CNN merupakan model deteksi objek pengembangan dari model Fast-RCNN dengan memanfaatkan jaringan proposal wilayah (RPN-*Region Proposal Network*) yang berbasis model CNN (*Convolutional Neural Network*) tampak pada Gambar 8. RPN dan Fast-RCNN digabungkan menjadi satu jaringan dengan berbagi fitur konvolusionalnya yang memungkinkan pengenalan proposal wilayah dengan waktu dan biaya komputasi yang rendah yang secara bersamaan memprediksi batasan objek dan skor objektivitas di setiap posisi. RPN dilatih *end-to-end* untuk menghasilkan proposal wilayah berkualitas tinggi, yang selanjutnya digunakan oleh Fast R-CNN untuk proses deteksi.



Gambar 8. Arsitektur Faster-RCNN (Ren *et al.*, 2017)

Pada penelitian (Laili, 2022) membuat sistem yang secara otomatis dapat menghitung orang yang masuk, keluar dan orang yang sedang berada di ruangan melalui kamera menggunakan metode Faster-RCNN, mendapatkan akurasi 100%, *recall* 1, *precision* 1 dan *F1-Score*-nya 1. Sedangkan penelitian lain yang menggunakan metode Faster-RCNN untuk mendeteksi pemakaian *masker* untuk prokes Covid-19 mempunyai akurasi 92,75% dan pendeteksian suhu tubuh menggunakan sensor MLX90614 mempunyai selisih rata-rata 0,18°C pada jarak 10 cm dibandingkan dengan pengukuran menggunakan *Thermal Gun* (Syahjaya, 2022).

Penerapan sebagai *mouse virtual* pada komputer (Hernando, 2023) menggunakan *framework* MediaPipe dan Faster-RCNN yang dapat mengontrol pergerakan *mouse*, fungsi tombol klik kiri (*enter*) dan klik kanan (*escape*) pada jarak 50 cm ukurannya 73,16%, jarak 20 cm (70,56%) serta pada jarak 1,4 m (62,6%).

3. HASIL DAN PEMBAHASAN

Pada sub-bab hasil dan pembahasan ini akan mengevaluasi performa dari metode MediaPipe, Faster-RCNN dan FPS dari sistem yang telah dibuat pada penelitian ini sebagaimana yang telah disebutkan di latar belakang yang ada di sub-bab pendahuluan. Sebagai contoh pembahasan dapat dilihat pada Gambar 7, pada Gambar 7 tersebut terlihat metode MediaPipe mampu mendeteksi *landmark keypoint* dari gestur jari tangan menggunakan kamera dari *handphone* melalui aplikasi Iriun yang terkoneksi secara *wireless* dengan *laptop* pada jarak 100 cm dalam keadaan pencahayaan ruang yang merata.



Gambar 7. Deteksi gestur jari tangan dengan metode MediaPipe

Parameter uji pada penelitian ini dilakukan pada 5 orang subyek, dimana setiap subyek memperagakan 4 bentuk gestur jari tangan ("0" [kipas mati], "1" [kipas kecepatan *level* 1], "2" [kipas kecepatan *level* 2], dan "3" [kipas kecepatan *level* 3]) dan setiap bentuk gestur tersebut diulang sebanyak 3 kali, sebagai contoh pada Tabel 1 menampilkan data hasil pengujian metode MediaPipe dengan jarak 50 cm antara pengguna atau subyek dengan *webcam*. Sebagaimana halnya seperti pada Tabel 1 maka semua pengujian tersebut juga dilakukan pada metode MediaPipe dan metode Faster-RCNN dengan pengambilan data pada jarak berbeda, yaitu: pada jarak antara subyek orang dengan *webcam* sejauh 10 cm, 50 cm, 100 cm dan jarak terjauhnya 175 cm.

Tabel 1. Pengujian metode MediaPipe pada jarak 50 cm

No	Jarak (cm)	Gestur	Hasil Deteksi		Akurasi (%)	Error (%)	FPS
			Benar	Salah			
1	50	0	8	2	80	20	24
	50	1	10	0	100	00	29
	50	2	9	1	90	10	23
	50	3	9	1	90	10	31
	50	0	1	9	10	90	30
	50	1	10	0	100	00	18
	50	2	9	1	90	10	22
	50	3	9	1	90	10	29
	50	0	8	2	80	20	18
	50	1	10	0	100	00	29
	50	2	10	0	100	00	17
	50	3	9	1	90	10	23
2	50	0	9	1	90	10	24
	50	1	9	1	90	10	22
	50	2	10	0	100	00	21
	50	3	10	0	100	00	26
	50	0	9	1	90	10	24
	50	1	9	1	90	10	18
	50	2	9	1	90	10	29
	50	3	9	1	90	10	19
	50	0	1	9	10	90	23
	50	1	9	1	90	10	18
	50	2	10	0	100	00	23
	50	3	9	1	90	10	21
3	50	0	8	2	80	20	25
	50	1	9	1	90	10	25
	50	2	8	2	80	20	25
	50	3	9	1	90	10	22
	50	0	9	1	90	10	24
	50	1	10	0	100	00	28
	50	2	9	1	90	10	31
	50	3	9	1	90	10	22
	50	0	9	1	90	10	23
	50	1	8	2	80	20	25
	50	2	9	1	90	10	19
	50	3	9	1	90	10	14
4	50	0	8	2	80	20	27
	50	1	10	0	100	00	28
	50	2	9	1	90	10	32
	50	3	8	2	80	20	23
	50	0	9	1	90	10	33
	50	1	9	1	90	10	23
	50	2	9	1	90	10	29
	50	3	9	1	90	10	28
	50	0	9	1	90	10	27
	50	1	9	1	90	10	32
	50	2	8	2	80	20	32
	50	3	8	2	80	20	32

No	Jarak (cm)	Gestur	Hasil Deteksi		Akurasi (%)	Error (%)	FPS
			Benar	Salah			
5	50	3	8	2	80	20	29
	50	0	8	2	80	20	23
	50	1	9	1	90	10	24
	50	2	9	1	90	10	22
	50	3	8	2	80	20	21
	50	0	1	9	10	90	29
	50	1	9	1	90	10	30
	50	2	8	2	80	20	24
	50	3	9	1	90	10	26
	50	0	8	2	80	20	31
	50	1	9	1	90	10	24
	50	2	9	1	90	10	19
	50	3	9	1	90	10	23
Rata – Rata Gestur Jari 0					70	30	25
Rata – Rata Gestur Jari 1					92,7	7,3	25
Rata – Rata Gestur Jari 2					90	10	24
Rata – Rata Gestur Jari 3					88,7	11,3	24

3.1. Hasil Pengujian Metode MediaPipe

Hasil pengujian deteksi gestur jari tangan untuk mengontrol *level* kecepatan putaran kipas angin dengan menggunakan metode MediaPipe pada jarak yang berbeda secara keseluruhan (10 cm, 50 cm, 100 cm dan 175 cm) dapat dilihat pada Tabel 2 di bawah. Berdasarkan data Tabel 2, dapat diketahui bahwasannya metode MediaPipe dalam mendeteksi gestur jari tangan memiliki akurasi tertinggi pada jarak 50 cm antara pengguna dan *webcam* dengan rata-rata akurasinya 85,35% dan rata-rata FPS 24,5.

Nilai rata-rata akurasi tertinggi kedua sebesar 71,68% didapatkan pada jarak 100 cm dengan nilai FPS 23,5 sedangkan pada jarak 10 cm antara pengguna dan *webcam* mempunyai tingkat akurasi rata-rata terendah, yaitu: 41,6% dengan nilai FPS 23, hal ini dikarenakan deteksi *keypoints* pada *landmarks* gestur jari tangan tidak maksimal. Spesifikasi dari jenis kamera yang digunakan juga menentukan tingkat kualitasnya sehingga memungkinkan terjadinya perubahan titik terdekat dan titik terjauh antara si pengguna dan kamera dalam mendeteksi gestur jari tangan.

Tabel 2. Pengujian metode MediaPipe pada semua jarak

Jarak (cm)	Gestur Jari	Akurasi (%)	Error (%)	FPS
10	0	37	63	24
	1	46,7	53,3	23
	2	42,7	57,3	23
	3	40	60	22
50	0	70	30	25
	1	92,7	7,3	25
	2	90	10	24
	3	88,7	11,3	24
100	0	54,7	45,3	24
	1	80	20	22
	2	78	22	24
	3	74	26	23
175	0	63,3	36,7	25
	1	80,7	19,3	25
	2	67,3	32,7	24
	3	66	34	23

3.2. Hasil Pengujian Metode Faster-RCNN

Tampilan pada Gambar 8 menunjukkan *region of interest* hasil komputasi deteksi gestur 3 jari tangan kanan dengan akurasi yang tampil di layar 75% melalui kamera *handphone* dengan menggunakan program Iriun *webcam* pada jarak 50 cm. Sedangkan pada Tabel 3 menampilkan data hasil pengujian menggunakan metode Faster-RCNN pada semua

jarak. Pada jarak 10 cm antara pengguna dengan *webcam* metode Faster-RCNN memiliki akurasi tertinggi dengan nilai rata-rata akurasi 36% dan FPS-nya 18,5. Sedangkan pada jarak 175 cm hasil deteksi gestur jari tangan pada metode Faster-RCNN mempunyai nilai akurasi rata-rata terendah, yaitu: 14,83% dengan FPS 18,25.



Gambar 8. Deteksi gestur jari tangan dengan metode Faster-RCNN

Tabel 3. Pengujian metode Faster-RCNN pada semua jarak

Jarak (cm)	Gestur Jari	Akurasi (%)	Error (%)	FPS
10	0	34	66	18
	1	60	40	24
	2	14	86	17
	3	16	84	17
50	0	24	76	21
	1	73	27	20
	2	14	86	20
	3	12	88	20
100	0	8,7	9,3	18
	1	58	42	21
	2	2,7	97,3	18
	3	5,3	94,7	17
175	0	4,7	95,3	17
	1	51,3	48,7	22
	2	2	98	17
	3	1,3	98,7	17

3.3. Pengecekan Komunikasi Data

Pengecekan komunikasi data perlu dilakukan untuk mengetahui berapa persen ketepatan proses pengiriman data melalui komunikasi data secara serial dari *laptop* yang menjalankan komputasi sistem deteksi gestur jari tangan dengan mikrokontroler Arduino Uno sebagai eksekutor *relay* penggerak *level* kecepatan putaran kipas secara otomatis sebagaimana tampak pada Gambar 9. Proses pengecekan ketepatan komunikasi data tersebut dilakukan pada jarak 100 cm antara pengguna dan *webcam* dengan pencahayaan ruangan yang merata.

Hasil dari proses pengecekan ketepatan komunikasi data dapat dilihat pada Tabel 4, pengambilan data dilakukan pada 5 orang yang mana tiap orang diulang sebanyak 3 kali dalam memperagakan 4 macam jenis gestur jari tangan yang mengontrol 4 *level* kecepatan putaran kipas angin (*level* 0, 1, 2 dan *level* 3).



Gambar 9. Pengecekan ketepatan komunikasi data

Tabel 4. Pengujian ketepatan pengiriman data

Subyek	Data Jari	Level Kecepatan Kipas				Hasil	
		0	1	2	3	Tepat	Salah
1	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
	0	x				x	x
	1		✓			✓	
	2			✓		✓	
	3				✓	x	x
2	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3		x			x	x
	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
3	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
	0	x				x	x
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
4	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
5	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
	0	✓				✓	
	1		✓			✓	
	2			✓		✓	
	3				✓	✓	
	0	✓				✓	
	1		✓			✓	
	2		x			x	x
	3			x		x	x
Jumlah komunikasi data						53	7

Berdasarkan pembacaan data pada Tabel 4 tersebut didapatkan hasil yang tepat antara bentuk

gestur jari tangan dengan data yang dikirim serta aksi yang dilakukan *relay* pada kipas angin sebanyak 53 kali dari jumlah data keseluruhan 60 kali maka persentase ketepatannya sebesar 88,33%.

4. KESIMPULAN

Berdasarkan pada bab hasil dan pembahasan serta eksperimen yang dilakukan secara empiris pada penelitian ini maka dapat disimpulkan beberapa hal, diantaranya:

1. Sistem kontrol *level* kecepatan kipas melalui deteksi gestur jari tangan menggunakan MediaPipe dan Faster-RCNN dapat diterapkan dengan baik serta sesuai harapan.
2. Akurasi rata-rata deteksi gestur jari tangan menggunakan MediaPipe pada jarak 10 cm (41,6%), jarak 50 cm (85,35%), jarak 100 cm (71,68%), dan pada jarak 175 cm sebesar 69,33%.
3. Akurasi klasifikasi rata-rata deteksi gestur jari tangan menggunakan Faster-RCNN pada jarak 10 cm (36%), jarak 50 cm (30,75%), jarak 100 cm (18,68 %), dan pada jarak 175 cm sebesar 14,83%.
4. Akurasi ketepatan komunikasi pengiriman data secara serial antara proses komputasi deteksi gestur jari tangan pada laptop dengan aksi kontrol *level* kecepatan kipas oleh *relay* yang dikendalikan Arduino Uno sebanyak 60 kali percobaan sebesar 88,33%.
5. Waktu komputasi metode MediaPipe pada jarak 10 cm, 50 cm, 100 cm dan 175 cm rata-rata nilai FPS-nya 23,75. Sedangkan Waktu komputasi metode Faster-RCNN pada jarak 10 cm, 50 cm, 100 cm dan 175 cm rata-rata nilai FPS-nya 19, maka dapat disimpulkan sistem kontrol *level* kecepatan kipas melalui deteksi gestur jari tangan pada kedua metode diatas dapat berjalan secara *realtime* (FPS antara 8-30).

DAFTAR PUSTAKA

- ALIYAH, A. N., 2022. *Implementasi Metode Human Activity Recognition (Har) Menggunakan Mediapipe Holistics Dan Algoritma Long Short Term Memory (Lstm) Untuk Menerjemahkan Gerakan Bahasa Isyarat Menjadi Kosa Kata*, 787. UIN Syarif Hidayatullah Jakarta. Available at: <https://repository.uinjkt.ac.id/dspace/handle/123456789/65450>.
- ARSAL, M., WARDIJONO, B. and ANGGRAINI, D., 2020. 'Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN', *Jurnal Nasional Teknologi dan Sistem Informasi*, 6, pp. 55–63. doi: 10.25077/TEKNOSI.v6i1.2020.55-63.
- ASMALENI, P., HAMDANI, D. and SAKTI, I., 2020. 'Pengembangan Sistem Kontrol Kipas Angin Dan Lampu Otomatis Berbasis Saklar Suara Menggunakan Arduino Uno', *Jurnal Kumparan Fisika*, 3(1), pp. 59–66. doi: 10.33369/jkf.3.1.59-66.
- AZIS, H. *et al.*, 2020. 'Performa Klasifikasi K-NN dan Cross Validation Pada Data Pasien Pengidap Penyakit Jantung', *ILKOM Jurnal Ilmiah*, 12(2), pp. 81–86. doi: 10.33096/ilkom.v12i2.507.81-86.
- 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/>.
- KURNAWAN, E., 2022. *Sistem Deteksi Range of Motion Pada Manusia Berbasis MediPipe Khusus Gerakan Flexion-Extension dan Abduction Adduction*. Universitas Muhammadiyah Yogyakarta. Available at: <https://etd.umsy.ac.id/id/eprint/34594>.
- LAILI, S. N., 2022. *TA : Sistem Deteksi Kapasitas Orang di dalam Ruangan Menggunakan Metode Faster R-CNN*. Universitas Dinamika. Available at: <https://repository.dinamika.ac.id/id/eprint/6315/>.
- MAKAHAUBE, S. S., SAMBUL, A. M. and SOMPIE, S. R., 2021. 'Implementation of Gesture Recognition Technology for Automated Education Service Kiosk', *Jurnal Teknik Informatika*, 16(4), pp. 1–8.
- 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/>.
- RAMADOSS, J. *et al.*, 2021. 'Computer Vision for Human-Computer Interaction Using Noninvasive Technology', *Scientific Programming*, 2021. doi: 10.1155/2021/3902030.
- REN, S. *et al.*, 2017. 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), pp. 1137–1149. doi: 10.1109/TPAMI.2016.2577031.
- SYAHJAYA, M. L., 2022. *TA: Sistem Deteksi Suhu Tubuh dan Pemakaian Masker pada Manusia menggunakan Metode Faster-RCNN secara Realtime*. Universitas Dinamika. Available at: <https://repository.dinamika.ac.id/id/eprint/6349/>.