

Pengenalan Gestur Jari Tangan Sebagai Media Pembelajaran Berhitung Bagi PAUD Berbasis Visi Komputer dan *Deep Learning*

Muhammad Rifki Pratama ¹, Heri Pratikno ^{2*}, Yosefine Triwidyastuti ³, dan Musayyanah ⁴

^{1,2,3,4} Teknik Komputer, Fakultas Teknologi dan Informatika, Universitas Dinamika, Indonesia

* Korespondensi: heri@dinamika.ac.id

Abstrak: Pendidikan anak usia dini (PAUD) merupakan periode perkembangan yang penting dalam kehidupan anak, salah satu pertumbuhan yang penting adalah aspek kognitif. Berhitung termasuk dalam kemampuan kognitif yang meliputi kemampuan menghafal, memahami, mengaplikasi, menganalisis, mensintesis, dan kemampuan mengevaluasi. Guna meningkatkan kemampuan berhitung bagi anak-anak maka harus diketahui bagaimana cara mengaplikasikan hitungan tersebut dalam dunia nyata. Pada penelitian ini memberikan solusi bagaimana cara belajar berhitung yang menyenangkan serta tidak membosankan bagi anak-anak menggunakan media pembelajaran berhitung yang interaktif melalui pengenalan bentuk gestur jari-jari kedua tangan berbasis visi komputer dan *deep learning* secara *realtime*. Adapun visi komputer pada penelitian ini menggunakan *framework* MediaPipe, sedangkan metode *deep learning* yang digunakan adalah arsitektur *Convolutional Neural Network* (CNN). Hasil pengujian yang telah dilakukan pada penelitian ini, akurasi deteksi kesepuluh jari tangan untuk proses berhitung dari metode MediaPipe sebesar 89,9% dengan *frame per second*-nya antara 20-25 FPS. Metode CNN persentase akurasi deteksinya 20% dengan nilai FPS-nya antara 10-12 FPS, manfaat dari hasil penelitian ini bisa memberikan pengalaman belajar yang menyenangkan, menstimulasi, dan mendukung perkembangan anak secara holistik.

Kata Kunci: CNN; Gestur Jari Tangan; Mediapipe; Media Pembelajaran; Visi Komputer

1. Pendahuluan

Kesuksesan pada masa depan dimulai dari pendidikan sejak usia dini, sehingga pendidikan sejak dini merupakan aspek penting dalam kehidupan seseorang. Undang-undang sistem pendidikan nasional Republik Indonesia No. 20 tahun 2003 pada Bab 1 Butir ke-14 menyatakan bahwa pendidikan sejak usia dini merupakan upaya pembinaan anak dari lahir hingga usia 6 tahun melalui pemberian stimulus pendidikan guna membantu tumbuh dan kembangnya jasmani dan rohani agar anak mempunyai kesiapan mengikuti pendidikan lebih lanjut melalui pembiasaan, memberikan dorongan, semangat dan keteladanan [1].

Pada masa pandemi COVID-19 di seluruh dunia pada umumnya dan Indonesia pada khususnya terjadi perubahan yang signifikan pada dunia pendidikan, dimana kegiatan proses belajar-mengajar dilakukan secara *daring* atau *online*. Perubahan tersebut mengakibatkan terjadinya *learning loss* terutama pada pendidikan anak usia dini yang memerlukan proses interaksi yang interaktif secara langsung belajar dengan guru serta bermain dengan teman sebaya. Salah satu pembelajaran yang terkena *learning loss* adalah pelajaran dasar berhitung pada anak PAUD. Terkait dengan permasalahan tersebut diatas maka diperlukan berbagai cara yang kreatif untuk memberikan solusi agar aktivitas belajar berhitung dapat dilakukan dengan menyenangkan serta tidak membosankan bagi anak-anak, salah satu metode yang dapat diterapkan serta mempunyai prospek yang besar pada bidang pendidikan adalah bidang kecerdasan buatan (*artificial intelligence*).

Beberapa penelitian-penelitian yang telah menerapkan *artificial intelligence* baik menggunakan metode *machine learning*, *deep learning* dan *computer vision* untuk proses deteksi gesture jari tangan antara lain: penelitian pengenalan gestur tangan berbasis penghitungan jari menggunakan CNN dan sensor *infra-red*, sebagai antarmuka alternatif pengganti fitur tombol sentuh konvensional di mobil dengan nilai akurasi 97% [2]. Penelitian [3] melakukan klasifikasi gestur tangan menggunakan arsitektur CNN untuk membaca bahasa isyarat SIBI dengan tingkat akurasi sebesar 88%. Deteksi gestur jari tangan menggunakan YOLO untuk pengendalian slide presentasi pada jarak 1 meter hingga 2,5 meter mendapatkan rata-rata akurasi 91,18% [4].

Pengenalan gestur tangan juga dapat digunakan untuk menggantikan fungsi *mouse* komputer melalui kamera dengan tingkat akurasinya 68% pada 75 kali percobaan [5]. Penelitian lain yang terkait dengan pengenalan gestur tangan untuk penerapan di layanan mandiri edukasi menggunakan kamera [6], disamping itu pengenalan gestur juga dapat digunakan sebagai alat bantu komunikasi pasien stroke berat yang terintegrasi dengan *internet of things* (IoT) untuk memberikan notifikasi perubahan gerak ke aplikasi berbasis Android [7].

Penulis pada penelitian ini menerapkan metode visi komputer dan *deep learning* yang digunakan sebagai media pembelajaran berhitung bagi anak PAUD melalui proses deteksi dan pengenalan gestur jari dari kedua tangan. Guna mengetahui seberapa *realtime* sistem yang telah dibuat maka juga dihitung FPS-nya (*frame per second*), dikatakan sistem berjalan dalam waktu nyata apabila *frame per second* nilainya diantara 8 – 30 FPS. Proses pengenalan gestur kesepuluh jari tangan secara visi komputer menggunakan *framework* MediaPipe sedangkan untuk metode *deep learning* menerapkan arsitektur CNN (*Convolutional Neural Network*), penjelasan dari kedua metode tersebut diatas dibahas pada *sub-bab* metode penelitian.

2. Bahan dan Metode

2.1. Perangkat Lunak

Pada sub-bab ini akan dibahas metode serta bahan-bahan pendukung utama penelitian, baik dari rancangan perangkat lunak maupun perangkat kerasnya. *Library* utama visi komputer pada penelitian ini menggunakan OpenCV (*Open Computer Vision*), OpenCV adalah *library open source* lisensi dari BSD-*Licensed* yang mempunyai lebih dari 2.500 macam algoritma yang mendukung keperluan penerapan visi komputer dan *machine learning*. Algoritma yang ada di OpenCV mempunyai fungsi dan kegunaan masing-masing, seperti halnya untuk proses deteksi dan pengenalan wajah, deteksi jenis kelamin, deteksi gerakan tangan, identifikasi objek dan sebagainya [8]. *Deep learning* merupakan metode pengembangan dari *machine learning* yang saat ini banyak digunakan dalam penelitian maupun penerapan kecerdasan buatan dalam aktivitas sehari-hari manusia, untuk mendapatkan *knowledge* maka diperlukan proses *training dataset* pada *deep learning*. *Deep learning* menjalankan perintah secara spesifik, perintah spesifik tersebut dapat digunakan untuk mendeteksi gambar, identifikasi gambar, klasifikasi gambar maupun untuk proses prediksi [9].

Bahasa pemrograman yang digunakan pada penelitian ini adalah bahasa pemrograman yang interpretatif, yaitu: Python, karena Python dapat memenuhi kebutuhan programming yang fokus pada tingkat keterbacaan kode serta banyak diterapkan pada bidang *machine learning*, *Internet of Things* dan *data science* [10]. CNN merupakan salah satu arsitektur dari *deep learning* yang digunakan untuk proses data dari suatu citra, sehingga mampu mengenali dan mendeteksi objek yang diinginkan dari suatu citra. CNN selain dapat melakukan proses klasifikasi gambar, juga dapat digunakan untuk memecahkan permasalahan pada bidang *Computer Vision* [11]. Perusahaan Google membangun sebuah *pipeline* yang digunakan untuk proses data persepsi dari berbagai format video dan audio dalam *framework* Mediapipe. Mediapipe dapat diterapkan pada rancangan aplikasi yang akan dibangun memerlukan kecerdasan buatan, *framework* dari Mediapipe memberikan sekumpulan solusi pada bidang *machine learning*, diantaranya: *iris detection*, *face detection*, *segmentation*, *holistic* dan sebagainya.

Jupyter Notebook mempunyai lingkungan komputasi interaktif yang populer sehingga memungkinkan pengguna untuk berbagi dan membuat dokumen untuk menggabungkan kode,

narasi dan visualisasi. *Jupyter* berbasis antarmuka *web* yang interaktif untuk pemrograman dari beberapa bahasa pemrograman, diantaranya Python, Julia dan R. *Jupyter Notebook* juga memiliki lingkungan yang fleksibel serta interaktif untuk eksplorasi data, pengembangan, komunikasi ilmiah dan pembelajaran mesin sehingga banyak digunakan oleh para peneliti, *data scientist*, komputasi jurnalisme dan *developer* untuk men-*develop* serta mendokumentasikan kode beserta proses analisis data.

2.2. Metode Penelitian

2.2.1 Instalasi *Environment* dan *Dataset*

Sebelum melakukan pemrograman untuk pengenalan gestur jari tangan berbasis visi komputer maupun pemrograman berbasis arsitektur *deep learning* maka diperlukan proses instalasi, setting dan konfigurasi *environment*. Pada proses *environment* melakukan instalasi semua *library* yang diperlukan agar program selanjutnya dapat berjalan dengan baik dan lancar sehingga tidak muncul pesan kesalahan yang ditampilkan pada saat program di-*running* atau di-*compile*. *Dataset* bentuk gestur jari tangan yang dipakai untuk proses *training* pada penelitian ini menggunakan *dataset* publik yang diambil dari alamat *web kaggle.com*, sebagaimana tampak pada Gambar 1 di bawah.

Proses augmentasi data juga dilakukan pada penelitian ini, augmentasi data merupakan salah satu cara untuk memperbanyak data latih yang dilakukan secara virtual dengan cara satu citra yang asli akan dikomputasi dengan dilakukan proses pengolahan citra di-*mirroring*, *zoom in/out*, *flip*, *rotate*, *crop* dan sebagainya. Proses augmentasi data tersebut tidak merubah arti atau makna citra aslinya, semakin banyak jumlah yang di-*training* maka akan semakin meningkatkan persentase akurasi dari *deep learning*. *Dataset* bentuk gestur jari tangan pada penelitian ini terdiri dari 11 jenis, yaitu: jari 0 sampai dengan jari 10, adapun *dataset* keseluruhan dari gestur jari tangan jumlahnya 21.600 citra yang mana setiap gestur jari mempunyai 1.800 citra dengan rincian 18.000 citra untuk proses *training* dan 3.600 citra gestur jari untuk proses validasi. Gambar 2(a) menampilkan *flowchart* proses *training* dan Gambar 2(b) menunjukkan *flowchart* proses validasi dari metode CNN.

Tahapan proses *training* yang dilakukan pada arsitektur CNN tampak pada Gambar 2(a), di mulai pada langkah *Start* hingga pengaksesan *dataset*, pada gambar 2(b) dari proses *training* dilakukan pengecekan kondisi, apakah proses *training* berhasil dilakukan jika belum maka dilanjutkan proses *training*nya, apabila sudah selesai maka program akan melanjutkan ke proses validasi *dataset*. Proses validasi akan melakukan tahapan *pre-testing*, hal ini dilakukan untuk mengetahui apakah *knowledge* atau pengetahuan dari CNN hasil dari proses *training* sudah mempunyai persentase *accuracy*, *specitivity* dan *sensitivity* yang tinggi, supaya pada saat dilakukan proses *testing* hasilnya tidak terjadi *overfitting* dan *underfitting*.

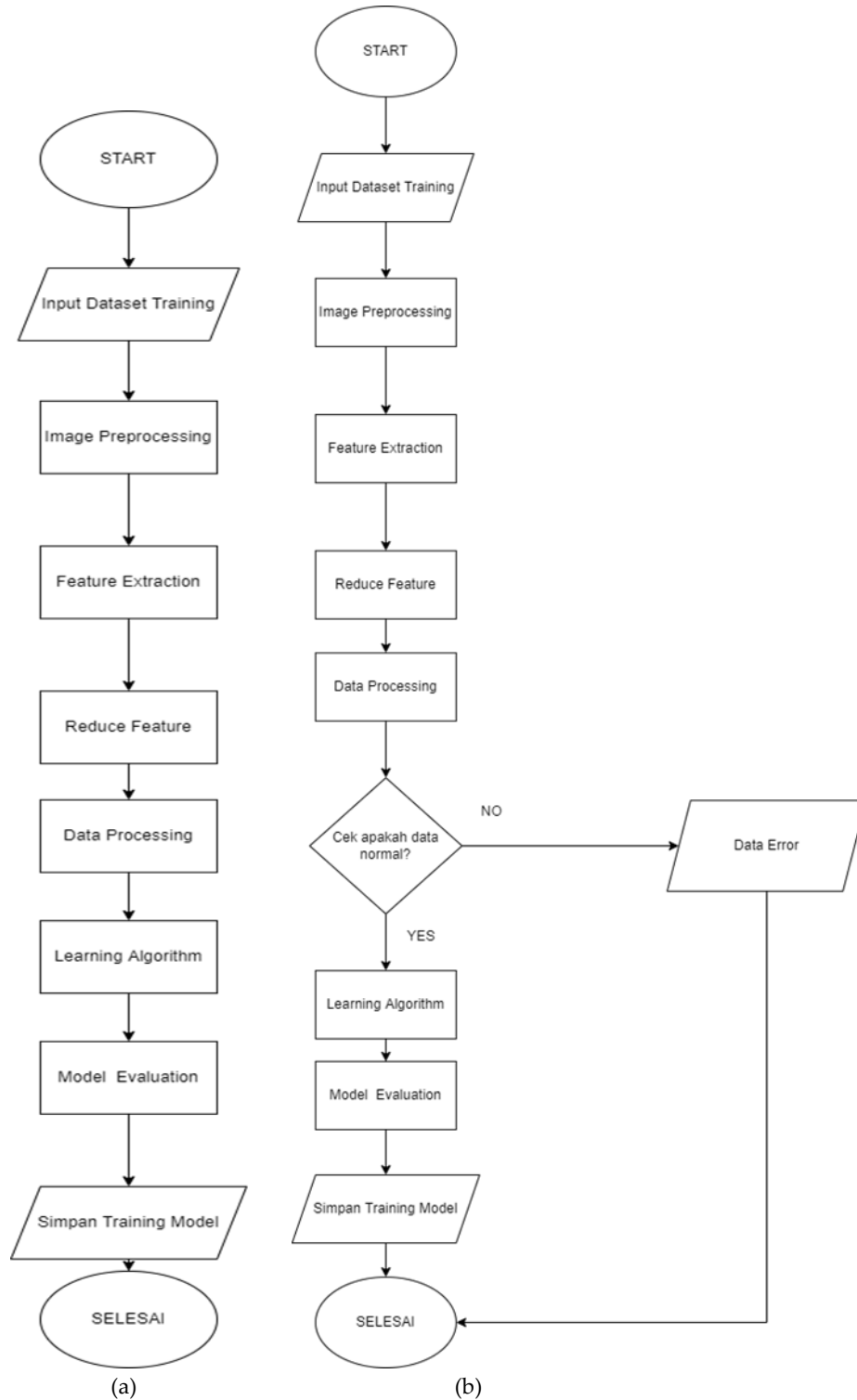


Gambar 1. *Dataset* gestur jari tangan

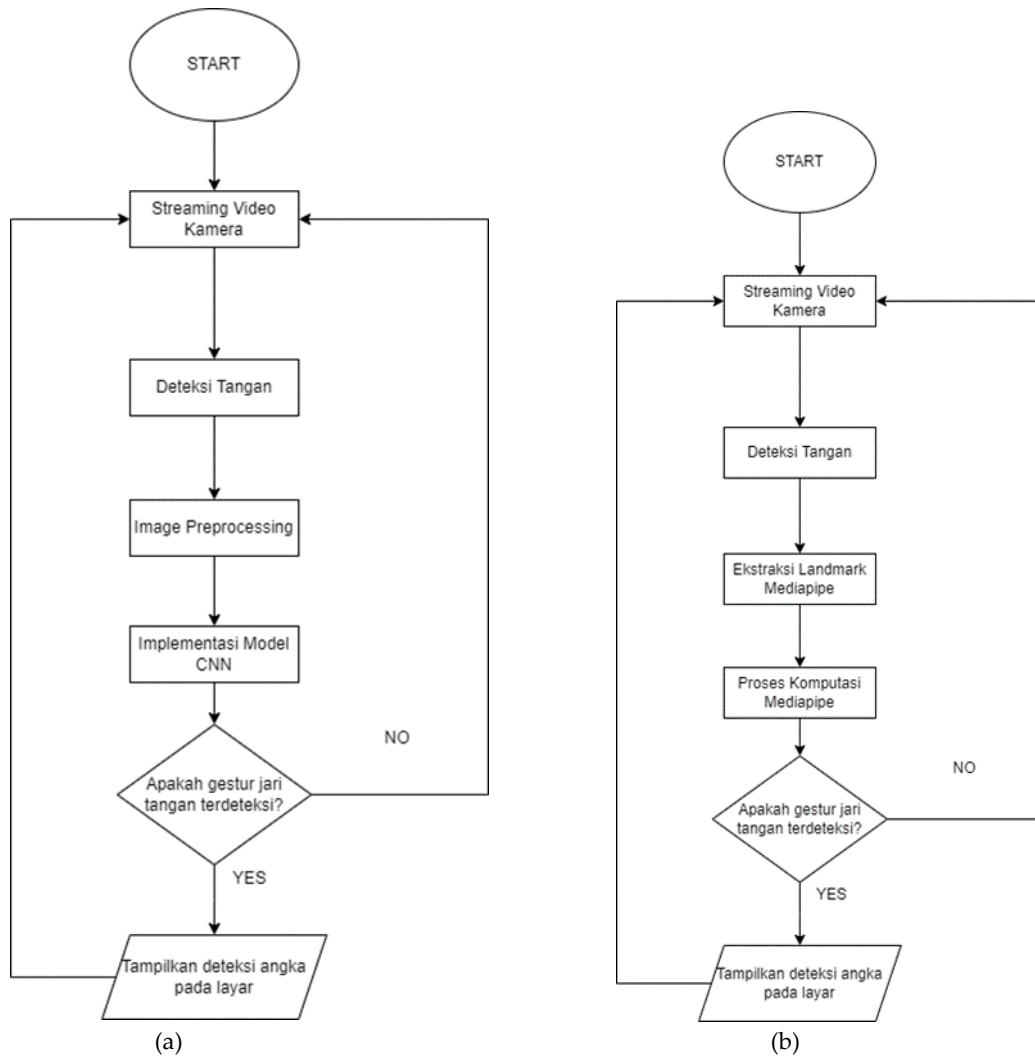
2.2.2 Metode Deteksi dengan CNN

Hasil dari proses *training* setelah dilakukan akan disimpan menjadi sebuah model, model ini akan dijalankan atau dipanggil pada saat CNN melakukan proses *testing*, hal ini dilakukan untuk mempersingkat waktu karena secara normalnya proses *training* ini akan perlu waktu yang cukup lama hingga berjam-jam tergantung dari besarnya dan banyaknya jumlah *dataset*-nya. Semakin banyak jumlah *dataset* maka waktu komputasinya juga akan semakin lama, guna mengetahui proses deteksi pada CNN dapat dilihat *flowchart*-nya pada Gambar 3(a). Pengujian proses deteksi menggunakan metode CNN dimulai dengan mengaktifkan *webcam*, jika terdeteksi gestur jari tangan

maka jari tangan akan ditampilkan di layar dengan diberi *squarebox* serta di layar akan ditampilkan angka yang menunjukkan jumlah dari jari tangan yang terdeteksi. Apabila jari tangan tidak terdeteksi maka di layar monitor tidak menampilkan apapun, program secara terus-menerus akan melakukan proses *scanning* atau *looping* secara terus-menerus untuk mendeteksi keberadaan dari jari tangan.



Gambar 2. Flowchart proses di CNN: (a) Proses training dan (b) Proses validasi



Gambar 3. Flowchart deteksi getur jari tangan: (a) Metode CNN dan (b) Metode MediPipe

2.2.3 Metode Deteksi dengan MediaPipe

MediaPipe merupakan aplikasi *open-source framework* lintas *platform* yang dapat memproses data-data multimedia, MediaPipe dikembangkan oleh Google untuk melakukan komputasi hasil dari proses *pre-trained* pada *dataset* skala besar yang didesain untuk melakukan berbagai tugas spesifik, diantaranya: estimasi pose, pengenalan wajah, deteksi objek, pelacakan tangan dan sebagainya. Bahasa dan *platform* pemrograman MediaPipe menggunakan *backend* TensorFlow sehingga dapat mengeksekusi CPU dan GPU secara efisien, MediaPipe dapat berjalan secara *realtime* di berbagai perangkat karena bersifat *open-source* maka pengembangan modelnya sangat pesat yang didorong oleh kolaborasi dari para komunitas. Proses deteksi menggunakan metode MediaPipe, jika ada gestur jari tangan yang tertangkap oleh kamera maka program akan menampilkan *landmark* yang fungsinya untuk melakukan klasifikasi gestur jari tangan. Adapun tampilan *flowchart* proses di MediaPipe dapat dilihat pada Gambar 3(b), apabila terdeteksi keberadaan jari tangan yang ditandai munculnya *landmark* maka hasil deteksinya akan ditampilkan di layar.

3. Hasil dan Pembahasan

Pada sub-bab ini akan dibahas hasil perbandingan kinerja dari kedua metode untuk mendeteksi dan mengenali gestur jari tangan, metode yang mana mempunyai akurasi paling tinggi dengan tingkat komputasi yang rendah, apakah metode yang berbasis visi komputer menggunakan MediaPipe atau metode yang berbasis *deep learning* menggunakan CNN. Proses pengujian dari kedua metode tersebut dilakukan dengan prosedur yang sama, yaitu: memberikan input deteksi yang sama, menjalankan proses deteksi dan pengenalan gestur jari tangan serta dilakukan perhitungan *frame per*

second (FPS). Hasil pembacaan pada Tabel 2, tampak rata-rata akurasi deteksi dari MediaPipe sebesar 89,9% lebih tinggi daripada akurasi dari CNN pada Tabel 1, *frame per second*-nya dapat mencapai 25 – 30 FPS, kelebihan dari metode MediaPipe ini didapat karena proses komputasinya lebih cepat dan lebih ringan daripada komputasi CNN, dalam proses deteksi gestur jari tangan MediPipe mempunyai fitur *landmark* sehingga hasil deteksinya lebih akurat walaupun dengan kontur *background* yang sama.

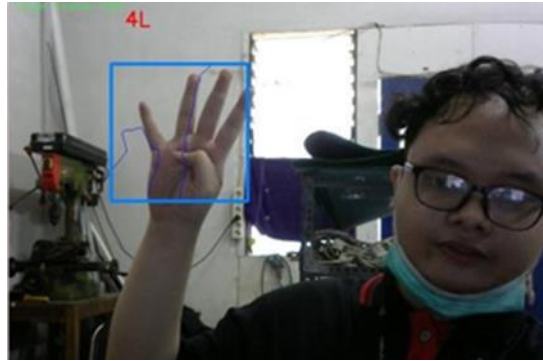
3.1. Hasil Pengujian pada Proses Deteksi

Proses pengujian serta pengambilan data pada penelitian ini dilakukan pada 5 orang dengan pengulangan sebanyak 3 kali dengan durasi 5 detik, 10 detik dan 15 detik. Setiap kali pengujian bentuk gestur jari kedua tangan memperagakan angka 0 sampai dengan angka 10, sehingga totalnya 11 buah. Jumlah pengambilan data setiap orang sebanyak 3 x 11 maka totalnya 33 bentuk gestur jari tangan, apabila hal tersebut dilakukan oleh 5 orang maka total data uji yang diambil sebanyak 165 buah. Perbandingan hasil pengujian proses deteksi dari metode CNN dan MediaPipe dapat dilihat pada Tabel 1 dan Tabel 2, pada Tabel 1 terlihat bahwa pengenalan gestur jari tangan menggunakan CNN rata-rata akurasinya masih rendah, yaitu: 20% dengan *frame per second*-nya antara 12 – 15 FPS, hal ini disebabkan komputasi dari metode CNN lebih berat daripada MediaPipe, disamping itu kondisi pencahayaan dan kontur *background* juga mempengaruhi. Pengujian proses deteksi tersebut menggunakan 11 gestur jari, yaitu: gestur jari 0 – 10, sebagai salah satu contoh gambar hasil dari proses deteksi perhitungan gestur jari angka 0 menggunakan MediaPipe terlihat pada Gambar 4 dengan kondisi semua jari tangan menggenggam atau mengepal sehingga jumlah semua nilai *keypoint* dari jari tersebut 0.

Salah satu contoh gambar hasil proses deteksi pengenalan gestur jari tangan menggunakan CNN terlihat pada Gambar 5, tampak CNN dapat mendeteksi 4 gestur jari tangan dan dikenali sebagai angka 4 yang tampil di layar. Proses deteksi dengan CNN menggunakan *region of interest* (ROI) untuk mendeteksi area dari objek yang diinginkan dan ekstraksi kontur yang didapatkan setelah objek dikonversi dari ruang warna RGB ke ruang warna BGR. Citra hasil dari konversi ruang warna tersebut akan dibandingkan dengan citra yang ada di *dataset* oleh model yang telah disimpan.

Tabel 1. Hasil Deteksi CNN

No	Subjek Uji	Percobaan ke-	Lama Uji (Detik)	Akurasi Deteksi Gestur Jari Tangan (%)											FPS
				0	1	2	3	4	5	6	7	8	9	10	
1.	Rifki	1	15	30	30	40	30	40	30	20	10	20	10	10	15
		2	10	30	40	40	30	30	10	20	10	20	10	10	16
		3	5	20	10	30	20	30	20	10	20	10	10	30	13
2.	Akbar	1	15	30	20	30	30	20	10	30	10	20	10	20	15
		2	10	30	30	30	20	10	30	10	20	10	10	20	17
		3	5	30	10	20	30	30	20	10	30	10	10	20	14
3.	Toriq	1	15	30	20	10	40	30	10	20	10	10	20	10	13
		2	10	30	20	30	10	30	20	10	20	10	10	20	16
		3	5	30	30	20	10	10	20	10	20	10	20	10	17
4.	Erwin	1	15	30	20	30	10	20	10	10	10	10	20	10	15
		2	10	10	20	10	30	20	30	10	20	10	20	20	12
		3	5	30	20	30	30	20	10	20	10	10	20	10	13
5.	Stifandy	1	15	30	30	20	30	30	20	10	10	20	10	20	14
		2	10	30	40	30	30	20	10	10	20	20	10	20	16
		3	5	30	20	40	30	30	20	20	10	10	20	10	12



Gambar 4. Deteksi bentuk gestur jari angka 4 menggunakan *landmark* CNN

Tabel 2. Hasil Deteksi MediaPipe

No	Subjek	Percobaan	Lama Uji Ke- (detik)	Akurasi Deteksi Gestur Jari Tangan (%)											FPS
				0	1	2	3	4	5	6	7	8	9	10	
1.	Rifki	1	15	90	90	80	100	90	90	90	80	90	90	90	26
		2	10	90	90	80	100	100	90	90	90	90	90	100	27
		3	5	90	100	90	100	90	90	100	90	90	90	100	30
2.	Akbar	1	15	80	100	90	100	90	90	90	100	90	80	100	28
		2	10	100	90	80	100	90	80	90	100	90	100	100	26
		3	5	100	90	90	80	100	90	100	90	90	80	90	27
3.	Toriq	1	15	100	90	100	90	80	90	90	100	90	90	90	26
		2	10	90	100	90	90	90	80	100	90	90	90	90	28
		3	5	90	90	100	90	90	80	80	90	90	90	100	30
4.	Erwin	1	15	90	90	90	100	100	90	90	80	90	90	100	25
		2	10	90	90	90	90	80	90	100	80	90	90	100	26
		3	5	90	100	90	90	90	80	80	90	90	90	100	27
5.	Stifa ndy	1	15	90	90	100	90	90	90	90	80	90	90	100	29
		2	10	90	90	100	90	100	90	100	90	90	90	100	28
		3	5	90	90	100	90	90	80	80	90	90	90	90	30



Gambar 5. Deteksi bentuk gestur jari angka 0 dengan *keypoints* menggunakan MediaPipe

3.2. Hasil Evaluasi Akurasi dan Loss

Berdasarkan data pada Tabel 3 dapat disimpulkan bahwa jumlah *epoch* pada saat proses *training* berkorelasi dengan peningkatan persentase akurasinya, model yang di-*training* akan mengalami

kenaikan *loss* pada bagian validasi pada saat *epoch* rendah, *validasi loss* terdeteksi stabil ketika *epoch* tinggi. kenaikan *loss* dapat disebabkan oleh beberapa faktor, salah satu diantaranya adalah kualitas input *dataset* yang kurang baik maka diperlukan tindakan awal *pre-processing*, *filtering* atau *image enhancement*. Jumlah *epoch* pada penelitian ini 5 *epoch*, berdasarkan eksperimen yang telah dilakukan pada penelitian ini, jika *epoch* lebih dari 5 maka akan terjadi kondisi *overfitting* atau *underfitting* yang akan mengakibatkan hasil pengukuran dari parameter uji tidak menggambarkan kondisi yang sebenarnya. Proses *training* dari 5 *epoch* tersebut memerlukan waktu total 17,113 detik dengan rata-rata waktu proses *epoch*-nya 3,423 detik.

Tabel 3. Hasil proses *training* dari CNN

Epoch	Lama Step	Loss	Accuracy	Validation Loss	Validation Accuracy
1	3.452 detik	0,5355	0,9254	10,1332	0.1667
2	3.439 detik	0,0565	0,9998	0,1292	0.9825
3	3.442 detik	0,0538	1,000	0,0538	1.000
4	3.414 detik	0,0527	1,000	0,0524	1.000
5	3.366 detik	0,0521	0,999	0,0512	1.000

4. Kesimpulan

Berdasarkan hasil dan pembahasan terkait dengan evaluasi performa dari MediaPipe dan CNN dalam mengenali gestur jari tangan sebagai media pembelajaran berhitung bagi anak PAUD didapatkan kesimpulan sebagai berikut: akurasi hasil *training* dari arsitektur *Convolutional Neural Network* sebesar 100% pada *epoch* ke-5 dengan total waktu komputasi selama 17,113 detik. Arsitektur *Convolutional Neural Network* memerlukan waktu komputasi 12 detik pada tiap *steps* dan 3,366 – 3,452 detik pada setiap *epoch*-nya. Hasil komparasi rata-rata nilai persentase akurasi dari kedua metode deteksi pada proses *testing* menunjukkan bahwa persentase akurasi *framework* Mediapipe mencapai 89,9% dengan FPS-nya 25-30 *frame per second*, sedangkan persentase rata-rata akurasi *Convolutional Neural Network* 20% dengan *frame per second*-nya 12-15 FPS. Performa metode arsitektur CNN kurang optimal untuk penerapan pada deteksi objek secara *realtime*, dikarenakan memerlukan proses *training dataset* gestur jari tangan sehingga proses komputasinya lebih berat daripada MediaPipe yang memang didesain untuk deteksi gestur tangan maka arsitektur CNN lebih cocok untuk proses klasifikasi citra. Penelitian ini yang berbasis metode MediaPipe dan CNN ke depannya akan dikembangkan serta dibandingkan kinerjanya dengan arsitektur *deep learning* yang lainnya, yaitu: menggunakan pre-trained network model LSTM, Faster-RCNN dan ResNet5.0.

Referensi

1. A. Nurhayati. "Konsep Pendidikan Anak Usia Dini Menurut Ki Hajar Dewantara," S1 thesis, Fakultas Ilmu Pendidikan, Universitas Pendidikan Indonesia, 2021.
2. S. Ahmed; F. Khan; A. Ghaffar; F. Hussain; S. H. Cho. "Finger-counting-based gesture recognition within cars using impulse radar with convolutional neural network," *Sensors (Switzerland)*, vol. 19, no. 6, 2019, doi: 10.3390/s19061429.
3. N. Anam. "Sistem Deteksi Simbol pada SIBI (Sistem Isyarat Bahasa Indonesia) Menggunakan Mediapipe dan Resnet-50," Undergraduate thesis, Universitas Dinamika, 2022.
4. M. R. Zaman; F. Utaminigrum. "Rancang Bangun Sistem Deteksi Gestur Tangan untuk Pengendalian Slide Presentasi menggunakan Algoritme You Only Look Once Versi 3," vol. 5, no. 8, pp. 3393–3398, 2021.

5. H. Yunita; E. Setyati. "Hand Gesture Recognition Sebagai Pengganti Mouse Komputer Menggunakan Kamera," *J. ELTIKOM*, vol. 3, no. 2, pp. 64–76, 2019, doi: 10.31961/eltikom.v3i2.114.
6. S. S. Makahaube; A. M. Sambul; S. R. Sompie. "Implementation of Gesture Recognition Technology for Automated Education Service Kiosk," *J. Tek. Inform.*, vol. 16, no. 4, pp. 1–8, 2021.
7. M. F. Nasution *et al.* "Perancangan Alat Bantu Komunikasi Pasien Stroke Berat Menggunakan Gesture Recognition Terintegrasi Internet Of Things Design a Communication Tool for Severe Stroke Patients Using Gesture Recognition Integrated with the Internet of Things," *Telekontran*, vol. 10, no. 2, pp. 128–138, 2022.
8. Andri Nugraha Ramdhon; Fadly Febriya. "Penerapan Face Recognition Pada Sistem Presensi," *J. Appl. Comput. Sci. Technol.*, vol. 2, no. 1, pp. 12–17, 2021, doi: 10.52158/jacost.v2i1.121.
9. M. Aarsal; B. Wardijono; D. Anggraini. "Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN," *J. Nas. Teknol. dan Sist. Inf.*, vol. 6, pp. 55–63, Jun. 2020, doi: 10.25077/TEKNOSI.v6i1.2020.55-63.
10. A. Zein. "Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON," *Sainstech J. Penelit. dan Pengkaj. Sains dan Teknol.*, vol. 28, Jul. 2018, doi: 10.37277/stch.v28i2.238.
11. A. Perdananto. "Penerapan deep learning pada Aplikasi prediksi penyakit Pneumonia berbasis Convolutional Neural networks," *J. Informatics Commun. Technol.*, vol. 1, pp. 1–10, Nov. 2019, doi: 10.52661/j_ict.v1i2.34.



© 2019 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

