

Implementasi *Virtual Mouse* Berbasis *Hand Gesture Recognition* dengan *MediaPipe* dan *Convolutional Neural Network*

Steven Suki¹, Chairisni Lubis^{*2}, Jeanny Pragantha³

^{1,2,3}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, Indonesia
Email: ¹steven.535210058@stu.untar.ac.id, ²chairisnil@fti.untar.ac.id, ³jeannyp@fti.untar.ac.id

Abstrak

Hand gesture recognition merupakan bentuk interaksi manusia-komputer yang inovatif dengan memanfaatkan gerakan tangan sebagai input. Penelitian ini bertujuan mengembangkan dan mengevaluasi aplikasi *virtual mouse* berbasis gerakan tangan untuk mengatasi keterbatasan perangkat input konvensional seperti masalah mobilitas dan ketergantungan hardware. Sistem dirancang menggunakan kombinasi *hand landmark detection* dengan *MediaPipe* untuk ekstraksi fitur tangan dan *Custom Convolutional Neural Network* (CNN) untuk mengenali pola gerakan. Hasil evaluasi menunjukkan performa model yang sangat baik dengan akurasi pelatihan mendekati 99% dan akurasi validasi mencapai 99,9%. Dalam pengujian aplikasi, sistem berhasil mengenali gestur dengan sempurna (100%) pada latar belakang putih, sedangkan pada latar belakang bervariasi tingkat keberhasilannya mencapai 66%. Temuan ini membuktikan bahwa solusi *virtual mouse* berbasis gerakan tangan dapat berfungsi efektif dalam kondisi terkendali, meskipun masih memiliki tantangan pada lingkungan dengan latar belakang kompleks.

Kata kunci: Convolutional Neural Network, Custom Convolutional Neural Network, Hand Gesture Recognition, Hand Landmark Detection, Interaksi Manusia Komputer.

Abstract

Hand gesture recognition offers an innovative approach to human-computer interaction by using hand movements as natural input. This research develops and evaluates a virtual mouse system based on hand gestures to address limitations of physical input devices such as restricted mobility and hardware dependence. The system utilizes *MediaPipe* for accurate hand landmark detection combined with a custom *Convolutional Neural Network* for precise gesture recognition. Experimental results show the model achieves high accuracy with 99 percent training accuracy and 99.9 percent validation accuracy. In practical testing, the application demonstrated perfect 100 percent gesture recognition on plain white backgrounds, while recognition rates decreased to 66 percent in environments with varied backgrounds. These findings indicate that the proposed vision-based virtual mouse performs reliably in controlled settings, though further improvements are needed for complex visual environments.

Keywords: Convolutional Neural Network; Custom Convolutional Neural Network; Hand Gesture Recognition; Hand Landmark Detection; Human Computer Interaction.

This work is an open access article and licensed under a Creative Commons Attribution-NonCommercial ShareAlike 4.0 International (CC BY-NC-SA 4.0)



1. PENDAHULUAN

Kemajuan Teknologi yang terus berkembang membuat peran teknologi sudah menjadi kebutuhan yang krusial. Interaksi Manusia Komputer (IMK) muncul karena kebutuhan manusia dalam berinteraksi dengan komputer dalam bentuk menu, teks, dan saat ini masih banyak sistem yang menggunakan perangkat input, salah satunya adalah mouse konvensional [1]. Penerapan interaksi manusia komputer pada perangkat keras seperti mouse dan keyboard sudah sangat umum, namun tidak untuk kamera web penerapan IMK menggunakan kamera web dapat menjadi banyak inovasi baru, salah satunya sebagai alat instruksi antar manusia dan komputer sebagai pengganti fungsi mouse, yang menerapkan *computer vision* yang merupakan cabang dari *artificial intelligence* [2]. *Artificial Intelligence* (AI) merupakan pecahan dari cabang ilmu computer yang melibatkan perhitungan matematika dan teknik pemrograman dalam menyelesaikan tugas yang biasanya dilakukan oleh manusia [3]. *Machine Learning* merupakan cabang dari kecerdasan buatan (AI) yang memungkinkan komputer atau sistem untuk mempelajari pola dari data historis dan mengambil keputusan secara cerdas berdasarkan pembelajaran tersebut [4]. *Machine Learning* masih membutuhkan fitur yang dihitung sebelum di latih, sedangkan *Deep Learning*

tidak memerlukan hal yang sama. *Neural Networks* merupakan model matematika yang perlu dilatih untuk melakukan tugas klasifikasi seperti otak manusia [5]. *Convolutional Neural Network* (CNN) merupakan arsitektur jaringan saraf tiruan yang dirancang khusus untuk mengolah dan mengenali pola dalam data berbasis citra atau gambar [6]. Computer Vision merupakan bidang yang berfokus pada pengembangan AI yang mampu memahami dan memprediksi informasi dari gambar atau data visual. Kemampuan ini memungkinkan teknologi seperti Virtual Mouse untuk mendeteksi gerakan dan pola tangan secara real-time, dan menjadikannya bagian penting dari teknologi tersebut [7]. Sekarang computer vision sudah semakin berkembang sehingga fitur yang lebih detail dalam deteksi bagian tubuh tertentu seperti *hand gesture recognition*. Deteksi tangan dilakukan dengan menggunakan *hand landmark detection* merupakan model yang dirancang secara akurat dalam menentukan posisi dari 21 titik kunci pada tangan dalam bentuk koordinat 2.5D.

Meskipun telah banyak penelitian yang mengembangkan sistem interaksi manusia dan komputer berbasis gestur tangan menggunakan teknologi *computer vision* dan CNN, sebagian besar fokus masih terbatas pada pengenalan gestur statis atau penggunaan *dataset* umum tanpa mempertimbangkan variasi gestur dinamis secara *real-time*. Selain itu, pendekatan yang ada sering kali belum mengintegrasikan secara optimal antara *hand landmark detection* untuk akuisisi fitur tangan dan arsitektur CNN untuk klasifikasi gestur, terutama dalam konteks aplikasi sebagai pengganti *mouse* konvensional.

Oleh karena itu, penelitian ini berkontribusi dengan merancang sistem virtual *mouse* berbasis *hand gesture* secara *real-time* menggunakan integrasi metode *hand landmark detection* untuk ekstraksi fitur dan *Convolutional Neural Network* untuk klasifikasi gesture. Sistem ini diharapkan mampu meningkatkan interaksi pengguna dengan komputer secara lebih alami, akurat, dan efisien, tanpa memerlukan perangkat input fisik seperti *mouse* atau *keyboard*.

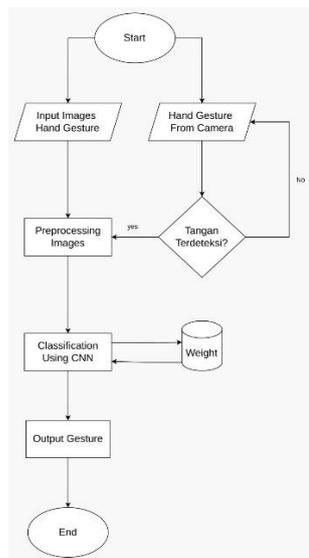
Penerapan ini bermanfaat untuk merancang sebuah sistem interaksi manusia komputer berbasis gesture yang inovatif serta bertujuan untuk meningkatkan efisiensi dan pengalaman yang lebih interaktif untuk mengontrol cursor mouse hanya dengan menggunakan gestur tangan yang dibuat menggunakan *hand landmark detection* untuk pendeteksian tangan dan metode *Convolutional Neural Network* (CNN) untuk pengenalan yang merupakan sebuah arsitektur yang terinspirasi dari biologi, CNN sendiri terdiri dari lapisan konvolusi, *pooling*, dan *fully-connected*, dan dapat dilatih secara efisien dengan pengawasan penuh [8].

2. METODE PENELITIAN

Rancangan penelitian virtual mouse dibuat menggunakan metode penelitian dengan pendekatan eksperimental. Pembuatan sistem dilakukan secara bertahap dengan perencanaan sistem, analisis sistem, pengambilan data, perancangan sistem, pengujian sistem, dan evaluasi sistem. Perencanaan merupakan hal pertama yang harus dilakukan agar pembuatan sistem dapat dirancang dengan baik. Tahapan ini dilakukan agar sistem dapat diimplementasikan sesuai dengan tujuan dari perencanaan yang dilakukan. Dengan perencanaan yang terstruktur, diharapkan dapat memberikan hasil yang optimal dan dapat mencapai tujuan utama dari rancangan sistem yang dibuat yaitu sebuah aplikasi virtual mouse dengan mendeteksi gestur tangan sehingga dapat mengontrol cursor. Analisis Sistem merupakan tahap yang berfokus pada tahap yang berfokus pada analisis data, pengolahan data, dan sumber referensi yang dipakai dan berupa tesis, jurnal, maupun karya ilmiah. Pengumpulan data dilakukan dengan menggunakan program yang sudah dirancang yang dapat mendeteksi tangan dan dikumpulkan menjadi 4 kelas yaitu moving cursor, mouse click, mouse drag, dan control volume. Perancangan dilakukan agar rancangan dapat terstruktur dengan baik dan dapat lebih mudah untuk mengidentifikasi potensi masalah jika ada. Tahap ini akan menjelaskan tujuan dari perancangan yang dibuat dan memberikan informasi mengenai rancangan yang akan dibuat agar bisa mendeteksi tangan yang terdapat pada kamera web desktop dan dapat memberikan informasi mengenai gestur yang dilakukan serta mengontrol kursor hanya dengan menggunakan gestur tangan untuk mengontrol mouse. Pengujian dilakukan dengan menggunakan uji blackbox dan uji dengan menggunakan parameter lapisan model untuk meneliti model yang paling cocok untuk digunakan. Evaluasi dilakukan untuk mengetahui performa dari model dan aplikasi yang sudah dibuat dengan melakukan analisis terhadap akurasi dan loss yang telah diuji.

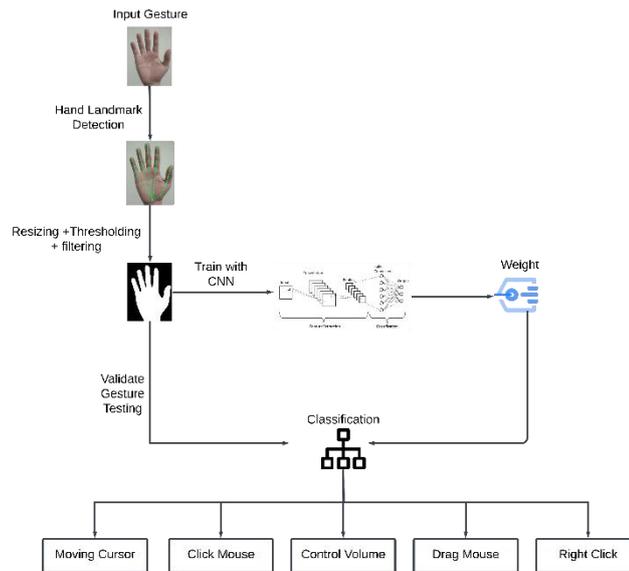
2.1. Perancangan Sistem

Rancangan sistem pada penelitian ini merupakan sebuah aplikasi *virtual mouse* menggunakan metode *Convolutional Neural Network* pada pelatihan pada data input yang akan menghasilkan nilai bobot yang akan dipakai dalam pengujian. Sistem yang dirancang terbagi menjadi 2 bagian yaitu pelatihan dan pengenalan. Pada sistem pelatihan input data dataset dengan bentuk gestur tangan yang berbeda sebanyak 5 kelas yaitu *moving cursor*, *mouse click*, *right click*, *mouse drag*, dan *control volume*. Data akan melalui tahap *resizing*, *preprocessing* dengan melakukan *thresholding* pada gambar lalu dinormalisasikan. Data yang sudah bersih akan dilanjutkan ke tahap deteksi *skeleton* menggunakan *Hand Landmark Detection*, lalu data akan dilatih menggunakan metode CNN dengan model *custom*. Setelah pelatihan selesai hasil yang didapatkan berupa bobot yang akan digunakan pada proses pengenalan.



Gambar 1 Flowchart Pelatihan dan Pengujian

Rancangan sistem yang dibuat merupakan aplikasi *virtual mouse* yang mengambil input gestur tangan dari kamera web. Sistem perancangan ini dirancang dan akan dilatih menggunakan metode *Convolutional Neural Network* (CNN) dengan model *Custom CNN*. Data pelatihan berupa dataset gambar gesture tangan yang sudah ditentukan memiliki 5 kelas. Data yang digunakan berjumlah 3095 untuk setiap kelas yang total 15475 gambar dan berukuran 300x300. Sebelum dilatih, data gambar akan diproses dengan beberapa tahapan *preprocessing*, yaitu diubah ke tipe grayscale dan dilakukan *resizing* menjadi 100x100 piksel agar seragam dan lebih efisien dalam proses pelatihan. Selanjutnya, dilakukan proses *thresholding* dengan nilai ambang batas 100–200 untuk memisahkan bentuk tangan dari latar belakang secara kontras, sehingga area tangan menjadi putih dan latar belakang menjadi hitam. Setelah *thresholding*, digunakan median blur dengan kernel sebesar 5 piksel untuk mengurangi noise dan memperhalus kontur gambar agar fitur tangan lebih bersih dan stabil saat dikenali oleh model. Data yang sudah bersih akan dilanjutkan ke proses pelatihan menggunakan model *Custom CNN*. Data dibagi menjadi 3 bagian yaitu pelatihan, pengujian dan validasi, data dibagi menjadi 80% untuk *training*, 10% untuk *testing*, dan 10% *validation*. Banyak data untuk *training* sebesar 12380 gambar, *testing* sebesar 1547 gambar, dan *validation* sebanyak 1238 gambar. Data dilatih dengan menggunakan *earlystopping* dengan memonitor pengurangan dari *validation loss* dengan *patience* sebesar 2 dan *optimizer adam* dengan *learning rate* sebesar 0,001 agar pelatihan dapat berhenti dalam *epoch* terbaik dengan *batch size* sebesar 32. Flowchart pelatihan dan pengujian dapat dilihat pada Gambar 1.



Gambar 2. Rancangan Sistem Virtual Mouse

Data pengujian akan diambil melalui kamera web secara *real time* dengan mendeteksi gestur tangan tertentu dengan menggunakan *hand landmark detection* dan membentuk *bounding box* yang dilakukan *preprocessing* juga untuk mendapatkan hasil gambar *crop* tangan yang terdeteksi sesuai ukuran *bounding box* lalu akan ditaruh kedalam *frame* berukuran 300x300 piksel. Dari data gambar yang sudah bersih akan dilakukan prediksi menggunakan model *Custom CNN* yang telah dibuat dan dilanjutkan pada tahap nantinya akan diklasifikasikan dengan nilai bobot dari hasil training dan akan memberikan prediksi bentuk gestur tangan sesuai dari bentuk yang terdeteksi. Terdapat 5 kelas *output* yaitu *moving cursor*, *mouse click*, *control volume*, *mouse drag*, dan *right click*. Gambar rancangan sistem virtual mouse dapat dilihat pada Gambar 2

2.1.1. Dataset Gestur Tangan

Tangan yang digerakan dapat menjadi beragam pose yang memiliki banyak makna tersendiri dalam kesepakatan umum maupun personal. Dengan melihat pose tangan dan menginterpretasikan makna dari pose tersebut dapat diperoleh informasi yang mudah dipahami. Sebagai bagian tubuh dari manusia tangan secara alami terbiasa digunakan sebagai media komunikasi, seperti saat bergerak tangan akan membentuk gestur tertentu [9]. Pada perancangan yang dibuat data gestur terdiri dari 5 kelas yaitu *moving cursor*, *mouse click*, *right click*, *drag mouse*, dan *control volume*. Dataset dikumpulkan secara manual dengan menggunakan kamera handphone yang telah diprogram untuk mendeteksi tangan. Tangan yang terdeteksi akan dilakukan *bounding box* disekeliling tangan. Pose tangan dilakukan sesuai dengan kelasnya lalu menekan tombol akan menyimpan hasil *crop* tangan sesuai pada folder kelasnya. Tombol m pada *keyboard* digunakan untuk menyimpan kelas *moving cursor*, c untuk *mouse click*, r untuk *right click*, d untuk *drag mouse*, dan v untuk *control volume*. Dataset yang berhasil terkumpul dengan total 15475 gambar yang terdiri dari setiap kelas sebanyak 3095 gambar dapat dilihat pada Gambar 3.



Gambar 3. Dataset 5 Kelas Gestur Tangan

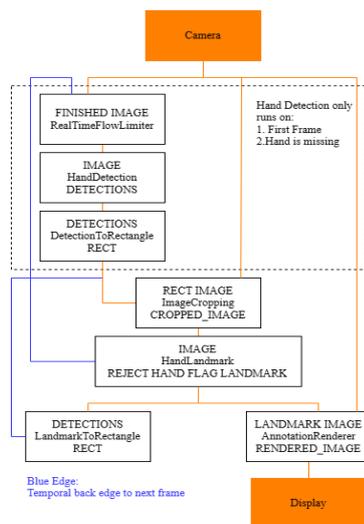
2.1.2. Thresholding

Thresholding merupakan tahap *preprocessing* pada gambar yang paling simpel untuk segmentasi gambar. *Thresholding* sering digunakan sebagai metode segmentasi area didasarkan pada lebih rendah atau lebih tinggi dari ambang *threshold*. Sekarang terdapat teknik bernama *threshold adaptive* [10].

$$g(x, y) = \begin{cases} 1, & \text{jika } f(x, y) \geq T \\ 0, & \text{jika } f(x, y) < T \end{cases} \quad (1)$$

2.1.3. Hand Landmark Detection

Saat ini ada banyak pendekatan yang bermunculan mengenai deteksi tanda pada tangan, yang biasa dipakai untuk ekstraksi ciri dalam memperkirakan pose tangan yang dapat berhasil diselesaikan dalam banyak kasus. Jika dikombinasikan dengan beberapa bentuk gestur tangan akan cukup untuk membedakan sejumlah pose tangan. Model *hand landmark* terbagi menjadi dua bagian yaitu pendeteksian tangan dan perhitungan landmark tangan [11]. Tahap awal dimulai dengan pendeteksian tangan pada video gambar yang ditangkap oleh kamera web. Kemudian tangan yang terdeteksi akan diberikan kotak disekeliling tangan (*Bounding Box*) yang berguna sebagai pembatas agar gambar dapat terpotong sesuai dari ukuran kotak tersebut. Dalam penerapan hand landmark dalam aplikasi yang dibuat dengan kombinasi titik landmark sebanyak 20 titik yang berada pada setiap ruas dan ujung jari pada tangan. Gambar grafik alur hand landmark detection dapat dilihat pada Gambar 4, dan posisi titik landmark pada Gambar 5.



Gambar 4. Hand Landmark Detection Graph [8]



Gambar 5. 21 Titik Landmark [12]

2.1.4. Convolutional Neural Network (CNN)

CNN merupakan arsitektur *deep learning* yang memanfaatkan jaringan saraf tiruan (*Neural Network*) dalam prosesnya. CNN sering digunakan dalam klasifikasi gambar yang akan diambil dari sebuah inputan gambar kemudian diproses dan diklasifikasikan. CNN terdiri dari jaringan saraf yang memiliki *weight*, *bias*, dan *activation function*. Arsitektur CNN mengintegrasikan tahap ekstraksi fitur dan klasifikasi gambar menjadi satu kesatuan CNN, sehingga menyederhanakan struktur sistem yang digunakan untuk deteksi dan pengenalan pada gambar. CNN didasarkan dari beberapa Lapisan yaitu Lapisan konvolusi, lapisan *pooling*, dan jaringan *fully connected*.

2.1.4.1. Lapisan Konvolusi

Pada lapisan konvolusi, dilakukan ekstraksi fitur pada gambar dengan menggunakan *kernel*, terdapat dua komponen penting lainnya yaitu *stride* yang berfungsi untuk mengatur jarak setiap

pergeseran yang dilakukan oleh *kernel* dan *padding* yang berfungsi untuk menambahkan nilai 0 pada setiap sisi data *input* [13]. Lapisan Konvolusi akan mengkombinasikan dua matriks yang berbeda dan terbentuk matriks baru. Lapisan konvolusi akan mengaplikasikan kernel data input yang akan terus bergeser sesuai dari *offset* yang memungkinkan pada gambar [14].

2.1.4.2. Lapisan Pooling

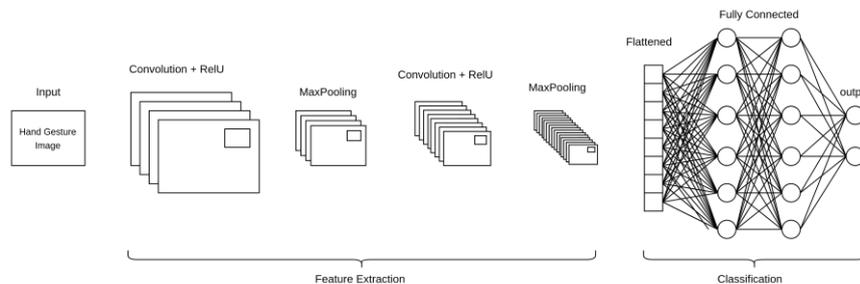
Pada lapisan *pooling* akan dilakukan reduksi dari hasil matriks konvolusi sebelumnya. Terdapat dua jenis *pooling* yang biasa sering digunakan dalam klasifikasi gambar yaitu *maximum pooling* dan *average pooling*. *Maximum pooling* melakukan reduksi dengan cara mengambil nilai terbesar dari nilai piksel yang masuk dalam ukuran kernel yang telah ditentukan. *Average pooling* melakukan reduksi dengan cara menjumlahkan semua nilai piksel yang masuk pada ukuran kernel dan menghitung rata-ratanya [13].

2.1.4.3. Lapisan Fully Connected

Pada tahap lapisan *fully connected* atau yang biasa lebih sering dikenal dengan lapisan *dense* dimana semua *neuron* dari lapisan sebelumnya akan terhubung semua dengan lapisan selanjutnya sama seperti jaringan syaraf tiruan [14]. Lapisan *fully connected* terdiri dari 3 lapisan lagi yaitu lapisan *input*, lapisan tersembunyi, dan lapisan *output*. *Input* dari lapisan *fully connected* berupa vektor yang didapatkan dari tahap *Flattening* yang berfungsi untuk mengubah matriks dua dimensi dari hasil ekstraksi fitur menjadi satu dimensi [13].

2.1.5. Model Custom CNN

Membangun arsitektur CNN merupakan pondasi untuk membuat sebuah model yang dapat memprediksi suatu hal. Dengan menggunakan *custom CNN* dapat lebih fleksibel mengatur berapa banyak lapisan yang akan digunakan untuk membuat suatu model. *Custom CNN* yang akan disusun terdiri dari ekstraksi fitur dan juga klasifikasi. Setelah model selesai dirancang maka akan siap digunakan dalam proses training [15]. Contoh gambar *custom CNN* dapat dilihat pada Gambar 6



Gambar 6. Custom CNN

2.2. Pengujian Sistem

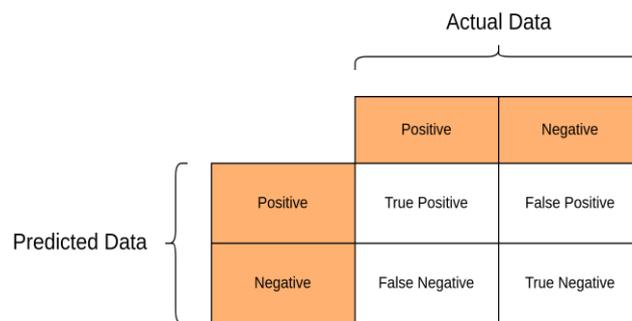
Pada pengujian sistem, dilakukan untuk mengetahui apakah aplikasi yang dirancang sudah sesuai dengan kebutuhan dan tujuan yang diharapkan yaitu dapat mengontrol dan mengklasifikasi pergerakan cursor hanya dengan menggunakan gestur tangan dengan beberapa cara yaitu dengan menggunakan *Blackbox User Acceptance Test*, pengujian dengan parameter layer konvolusi dan *hidden*, dan pengujian keberhasilan aplikasi *virtual mouse*. Pengujian dimulai dengan mengarahkan tangan ke kamera web untuk mendeteksi gestur. Beberapa gestur tangan akan diuji, seperti gerakan untuk menggerakkan kursor, klik, drag, dan control volume, dan right click. Setiap gestur akan diamati apakah aplikasi mampu mengenali dan menerjemahkannya menjadi perintah kursor sesuai yang diharapkan. Pengujian blackbox dilakukan dengan 3 orang penguji. Pengujian system dilakukan dengan perangkat dengan spesifikasi perangkat yang dapat dilihat pada Tabel 1.

Tabel 1. Spesifikasi Perangkat Sistem

Kategori	Komponen	Spesifikasi
Perangkat Keras	Prosesor	Intel i7 – 1065G7
	RAM	16 GigaByte
	Penyimpanan	256 GigaByte
Perangkat Lunak	Sistem Operasi	Ubuntu
	Bahasa Pemograman	Python
	Code Editor	Visual studio Code
	Software	Jupyter Notebook

2.3. Evaluasi Sistem

Pada evaluasi sistem, terdapat dua macam evaluasi yang dilakukan dalam pembuatan aplikasi virtual mouse ini yaitu *confusion matrix* dan *clasification report*. Penerapan *confusion matrix* digunakan untuk membuat visualisasi terhadap performa model yang digunakan terdapat . *Clasification report* dapat memberikan catatan informasi terhadap *accuracy*, *precision*, *recall*, dan *f-1 score* untuk setiap kelas. Bentuk *confusion matrix* dapat dilihat pada Gambar 7.



Gambar 7. *Confusion Matrix*

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (2)$$

$$Precision = \frac{(TP)}{(TP + FP)} \quad (3)$$

$$Recall = \frac{(TP)}{(TP + FN)} \quad (4)$$

$$F1Score = \frac{2 * (Recall * Precision)}{(Recall + Precision)} \quad (5)$$

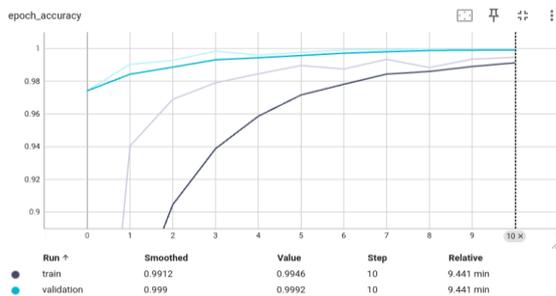
3. HASIL DAN PEMBAHASAN

3.1. Hasil Training

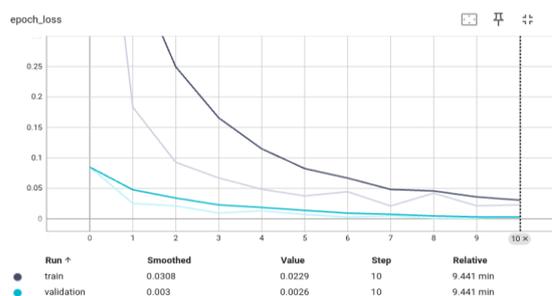
Hasil training yang didapatkan menunjukkan bahwa proses pengenalan gestur tangan pada aplikasi *virtual mouse* ini dapat dikenali dengan baik dan akurat. Model *custom CNN* terbaik didapatkan dengan melakukan pengujian pengaruh banyak lapisan konvolusi, *maxpooling*, dan tersembunyi terhadap model yang bertujuan untuk mendapatkan akurasi prediksi model CNN tertentu dengan mengevaluasi metrik performa seperti akurasi dan *loss* dari pelatihan maupun validasi. Hasil training dapat dilihat pada Tabel 2 dan grafil akurasi dan loss dapat dilihat pada Gambar 8 dan Gambar 9.

Tabel 2. Pengujian Parameter Lapisan Konvolusi dan Tersembunyi

Lapisan Konvolusi	Lapisan Maxpooling	Lapisan Tersembunyi	Akurasi Training	Loss Training	Akurasi Validasi	Loss Validasi
1	1	2	96.53%	9.93%	99,53%	1,48%
2	2	2	98.58%	4.35%	99.78%	1.11%
3	3	2	98.65%	4.2%	99.83%	0.56%
4	4	2	99.12%	3.08%	99.9%	0.3%
4	4	3	97.21%	7.6%	99.79%	1.15%
4	4	4	19.57%	160.97%	19.31%	160.96%



Gambar 8. Grafik akurasi *training custom CNN*



Gambar 9. Grafik *loss training custom CNN*

Pengujian diatas dilakukan dengan menggunakan *earlystopping* agar semua nilai akurasi yang didapatkan sudah merupakan *epoch* terbaik dengan menggunakan *earlystopping* pada setiap model. Hasil dari pengujian memperlihatkan akurasi yang semakin terus meningkat rata-rata 0.8% dan validasi sebesar 0.12% setiap lapisan konvolusi bertambah, dan penurunan loss rata-rata 2.28% dan validasi loss sebesar 0.39%. Pada parameter lapisan tersembunyi terdapat penurunan yang signifikan dari akurasi sebesar 39% setiap lapisan tersembunyi bertambah. Pengujian dengan hasil paling bagus didapatkan dengan menerapkan 4 lapisan konvolusi dan 2 lapisan tersembunyi dengan akurasi sebesar 99.12% bahkan saat pengujian dapat meningkat sampai 100%. Hasil Evaluasi model *Confusion Matrix* dapat dilihat pada Tabel 3.

Tabel 3. Evaluasi Model Confusion Matrix

Actual/predict	Control Volume	Drag Mouse	Mouse Click	Moving Cursor	Right Click
Control Volume	613	1	0	0	0

Actual/predict	Control Volume	Drag Mouse	Mouse Click	Moving Cursor	Right Click
Drag Mouse	0	595	0	0	0
Mouse Click	0	0	603	0	0
Moving Cursor	1	0	0	644	1
Right Click	0	0	0	0	637

Pengujian laporan klasifikasi yang dilakukan bertujuan agar performa model dapat dinilai secara menyeluruh dalam bentuk metrik evaluasi seperti *recall*, *precision*, dan *f1-score* pada setiap gestur yang dilabelkan. Laporan klasifikasi ini dapat membantu mengidentifikasi kelas yang mungkin memerlukan penyesuaian model yang lebih baik ataupun penambahan data agar akurasi semakin tinggi. Hasil evaluasi model klasifikasi dapat dilihat pada Tabel 4

Tabel 4. Evaluasi Model Laporan Klasifikasi

No.	Gestur Tangan	Precision	Recall	F1-Score	Akurasi
1.	Control Volume	99.84%	99%	99.92%	99.84%
2.	Drag Mouse	100%	100%	100%	100%
3.	Mouse Click	100%	100%	100%	100%
4.	Moving Cursor	99.69%	99.84%	99.76%	99.69%
5.	Right Click	99.84%	100%	99.92%	99.84%

3.2. Pengujian *Blackbox User Acceptance Test*

Pengujian *Blackbox UAT* dilakukan untuk memastikan bahwa aplikasi virtual mouse berfungsi sesuai dengan spesifikasi yang telah ditetapkan. Pengujian ini mengevaluasi kesesuaian antara input dan output aplikasi dengan menetapkan uji kasus [16]. Hasil pengujian *blackbox* dapat dilihat pada Tabel 5.

Tabel 5. Pengujian *Blackbox User Acceptance Test*

No.	Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Memulai Aplikasi	Klik Aplikasi Virtual Mouse	Aplikasi berhasil terbuka	Sesuai dengan ketentuan	Valid
2.	Modul Rekognisi Tangan	Klik Tombol Modul Rekognisi Tangan	Menampilkan Modul Rekognisi Tangan	Sesuai dengan ketentuan	Valid
3.	Pengenalan Gestur	Menggerakkan tangan sesuai gestur	Memberikan output informasi gestur tangan	Sesuai dengan ketentuan	Valid
4.	Kontrol kursor	Menggerakkan tangan sesuai gestur	Kursor bergerak sesuai dengan pergerakan tangan	Sesuai dengan ketentuan	Valid
5.	Control volume	Menggerakkan tangan sesuai gestur control volume	Volume pada desktop dapat diubah sesuai dengan tangan	Sesuai dengan ketentuan	Valid

6.	Modul Petunjuk Gestur	Klik Modul Petunjuk Gestur	Menampilkan Modul Petunjuk Gestur	Sesuai dengan ketentuan	Valid
7.	Modul Tentang	Klik Modul Tentang	Menampilkan Modul Tentang	Sesuai dengan ketentuan	Valid
8.	Tombol Kembali	Klik Tombol Kembali	Kembali ke Tampilan Awal	Sesuai dengan ketentuan	Valid

3.3. Hasil Pengujian Skenario

Pengujian skenario dilakukan untuk mengetahui seberapa baik aplikasi *virtual mouse* dapat digunakan saat digunakan langsung oleh pengguna. Setiap skenario diuji melalui 10 kali percobaan dengan latar belakang putih dan bercarasi, serta perbedaan pencahayaan. Hasil pengujian menunjukkan bahwa pada latar belakang putih, setiap gestur berhasil dikontrol dengan akurasi rata-rata sebesar 100%. Namun, pada latar belakang kotor, rata-rata keberhasilan menurun menjadi 66%. Secara keseluruhan, aplikasi *virtual mouse* mencapai rata-rata keberhasilan sebesar 83% dalam pengujian skenario ini. Hasil pengujian skenario pada latar belakang dapat dilihat pada Tabel 6.

Tabel 6. Pengujian Skenario Latar Belakang

No	Gestur	Jumlah Percobaan	Latar Belakang Pengujian			
			Bersih		Kotor	
			Benar	Salah	Benar	Salah
1.	Moving Cursor	10	10	0	4	6
2.	Mouse Click	10	10	0	7	3
3.	Right Click	10	10	0	3	7
4.	Drag Mouse	10	10	0	9	1
5.	Control Volume	10	10	0	10	0
Rata-rata Keberhasilan Latar Belakang			100%		66%	
Total rata-rata Keberhasilan			83%			

3.4. Hasil Pengujian Skenario

Hasil pengujian menunjukkan bahwa setiap penambahan lapisan konvolusi akurasi akan meningkat sebesar 0,8% dan mengurangi nilai *loss* sebesar 2,28%. Sebaliknya, penambahan lapisan tersembunyi cenderung menurunkan akurasi secara signifikan hingga 39%.

Model terbaik didapatkan dengan 4 lapisan konvolusi dan 2 lapisan tersembunyi mencapai akurasi sebesar 99,12%, *loss* sebesar 3,08%, akurasi validasi sebesar 99,9%, dan *loss* validasi sebesar 0,3%. Model ini diuji menggunakan ukuran batch sebesar 32 dan menggunakan *earlystopping* untuk batas *epoch*. Pengujian menghasilkan *confusion matrix* dengan tingkat kesalahan hanya berkisar 0,097%.

Laporan klasifikasi meliputi metrik *precision*, *recall*, *F1-score*, dan akurasi yang dihitung berdasarkan *confusion matrix* yang didapatkan. Pada kelas *control volume*, akurasi mencapai 99,84%; *drag mouse* dan *mouse click* masing-masing mencapai 100%; *moving mouse* mencapai 99,69%; dan *right click* mencapai 99,84%. Hasil ini menunjukkan bahwa model yang digunakan memiliki performa yang sangat baik dalam mengklasifikasikan setiap kelas.

Setelah mendapatkan model terbaik metode *blackbox* untuk memastikan bahwa semua fungsionalitas aplikasi *virtual mouse* berjalan dengan valid. Setelah itu, dilakukan pengujian secara *real-time* pada lingkungan dengan latar belakang putih dan berwarna serta variasi pencahayaan melalui 10

kali percobaan untuk setiap skenario. Hasil yang didapatkan menunjukkan bahwa pada latar belakang putih, aplikasi mencapai rata-rata keberhasilan sebesar 100%, sementara pada latar belakang kotor rata-rata keberhasilan menurun menjadi 66%. Total rata-rata keberhasilan aplikasi virtual mouse dari keseluruhan pengujian adalah 83%.

4. KESIMPULAN

Secara keseluruhan, aplikasi virtual mouse telah berfungsi sesuai dengan spesifikasi fungsionalitas yang diharapkan. Hasil terbaik pengujian parameter diperoleh pada model dengan 4 lapisan konvolusi dan 2 lapisan tersembunyi, yang mencapai akurasi sebesar 99,12% dan akurasi validasi sebesar 99,9%. Evaluasi performa model menggunakan confusion matrix menunjukkan hasil yang sangat baik, dengan tingkat kesalahan hanya sekitar 0,8%. Laporan klasifikasi memberikan metrik precision, recall, F1-score, dan akurasi yang mengesankan untuk setiap gestur. Hasil pengujian menunjukkan bahwa kelas control volume memiliki akurasi sebesar 99,84%, drag mouse dan mouse click masing-masing mencapai 100%, moving mouse mencapai 99,69%, dan right click sebesar 99,86%. Tingkat akurasi yang tinggi ini menunjukkan kemampuan aplikasi dalam mengenali gestur tangan secara akurat untuk mengontrol kursor. Namun, aplikasi menghadapi tantangan pada kondisi latar belakang yang bervariasi, seperti pencahayaan, warna, dan lokasi, yang menyebabkan penurunan akurasi sebesar 30-40%. Pada pengujian skenario, rata-rata keberhasilan mencapai 100% untuk latar belakang putih, tetapi menurun menjadi 60% pada latar belakang bervariasi. Dengan 10 kali percobaan untuk setiap skenario, rata-rata total keberhasilan aplikasi adalah 80%. Hasil ini menunjukkan bahwa meskipun aplikasi telah memiliki performa yang sangat baik, peningkatan akurasi pada kondisi latar belakang yang kompleks masih diperlukan untuk mencapai kinerja yang lebih konsisten. Terdapat beberapa saran yang dapat dikembangkan oleh peneliti berikutnya seperti Memperbanyak dataset pose tangan dengan latar belakang yang lebih bervariasi seperti pencahayaan, jarak, dan tempat dengan warna yang cukup kompleks, Penerapan bentuk gestur tangan baru yang lebih mudah untuk dibedakan dengan gestur lainnya agar memudahkan saat melakukan pengenalan, Menerapkan sistem drag untuk nilai thresholding pada interface aplikasi agar mempermudah deteksi tangan dan dapat mengatur proses thresholding yang lebih baik terhadap latar belakang yang terdeteksi, dan Pengembangan model custom CNN yang dapat lebih optimal lagi dengan melakukan pengujian dengan parameter yang berbeda-beda agar mendapatkan model yang lebih efisien dan akurat.

DAFTAR PUSTAKA

- [1] G. A. Rafiandy, "Laporan Tugas Akhir Teknik Komputer dan Jaringan Politeknik Negeri Jakarta," Politeknik Negeri Jakarta, Jakarta, 2022.
- [2] M. Z. Hapiz, "Implementasi Hand Gesture Recognition Pada Virtual Mouse Menggunakan Pustaka OpenCV," 2023.
- [3] M. Rizki dan E. Novianto, "Auditing Artificial Intelligence Menggunakan COBIT 2019," *Jurnal Sistem Informasi dan Informatika (JUSIFOR)*, vol. 2, no. 1, 2023, doi: 10.33379/jusifor.v2i1.1847.
- [4] R. Diana, H. Warni, dan T. Sutabri, "Penggunaan Teknologi Machine Learning Untuk Pelayanan Monitoring Kegiatan Belajar Mengajar Pada SMK Bina Sriwijaya Palembang," *JUTEKIN (Jurnal Teknik Informatika)*, vol. 11, no. 1, 2023, doi: 10.51530/jutekin.v11i1.709.
- [5] H. Joshi, N. waybhave, R. Litoriya, dan D. Mangal, "Design of a Virtual Mouse Using Gesture Recognition and Machine Learning," 1 Juni 2022, *Research Square*. doi: 10.21203/rs.3.rs-1616375/v2.
- [6] C. R. Wairata, E. R. Swedia, dan M. Cahyanti, "Pengklasifikasian Genre Musik Indonesia Menggunakan Convolutional Neural Network," *Sebatik*, vol. 25, no. 1, 2021, doi: 10.46984/sebatik.v25i1.1286.
- [7] R. Shinganjude, D. Majrikar, A. Bawiskar, S. Bobhate, J. Chaurawar, dan G. Ninawe, "Virtual Mouse Using AI and Computer Vision," *Int J Innov Sci Res Technol*, vol. 8, no. 11, hlm. 327–328, Nov 2023, doi: 10.5281/zenodo.10158688.
- [8] M. Arslan, "CNN Based Head-Mouse Control System for People With Disabilities a Thesis Submitted to the Institute of Graduate Studies of Near East University," Near East University, Nicosia, 2021.

- [9] A. A. Wibowo dan D. Astuti, “Gestur Tangan Manusia Dalam Karya Fotografi Seni,” vol. 18, no. 1, hlm. 114, Okt 2021.
- [10] S. Jardim, J. António, dan C. Mora, “Image thresholding approaches for medical image segmentation-short literature review,” dalam *Procedia Computer Science*, Elsevier B.V., 2023, hlm. 1485–1492. doi: 10.1016/j.procs.2023.01.439.
- [11] T. Grzejszczak, M. Kawulok, dan A. Galuszka, “Hand landmarks detection and localization in color images,” *Multimed Tools Appl*, vol. 75, no. 23, hlm. 16370–16371, Des 2016, doi: 10.1007/s11042-015-2934-5.
- [12] “Hand landmarks detection guide | Google AI Edge | Google AI for Developers.” Diakses: 14 Desember 2024. [Daring]. Tersedia pada: https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker
- [13] Tiffany, “Pengenalan Karakter Mandarin Menggunakan Metode Convolutional Neural Network dengan Model VGG-19,” Universitas Tarumanagara, 2024.
- [14] K. Azmi, S. Defit, dan Sumijan, “Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat,” vol. 16, no. 1, hlm. 33–34, Jun 2023, doi: 10.52072/unitek.v16i1.504.
- [15] K. Yogeswara Rao dan K. Srinivasa Rao, “Breast Cancer Image Classification Using Custom CNN,” *J Theor Appl Inf Technol*, vol. 31, no. 6, hlm. 2499, Mar 2024, [Daring]. Tersedia pada: www.jatit.org
- [16] R. Setiawan, “Black Box Testing Untuk Menguji Perangkat Lunak - Dicoding Blog.” Diakses: 14 Desember 2024. [Daring]. Tersedia pada: <https://www.dicoding.com/blog/black-box-testing/>