

Rancang Bangun Sistem Virtual Mouse Dengan Hand Gesture Recognition Menggunakan Convolutional Neural Network

Jeffrey Ken¹, Tony Darmanto², Hendro³

Informatika Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak

¹20421339_jeffrey_k@widyadharma.ac.id, ²tony.darmanto@yahoo.com, ³hendro@widyadharma.ac.id

Abstract

The development of human-computer interaction technology continues to grow, especially in the context of hand gesture recognition for device control applications. One interesting application is designing a virtual mouse system based on hand gesture recognition using a Convolutional Neural Network (CNN). This research was conducted to overcome the limitations of conventional input devices and utilize modern technology to create a more sophisticated and adaptive control system. This research focuses on developing a system that is able to identify and interpret hand gestures accurately in a real-time environment. The proposed system utilizes an RGB camera to capture images of the user's hand gestures, which are then processed by a CNN that has been pre-trained to recognize different gestures. Experiments were carried out using a gesture dataset that included variations in pose and hand orientation to evaluate the system's performance in recognizing gestures with a sufficient level of accuracy. The research results show that the system can successfully control the movement of the cursor on the computer screen with high accuracy, validating the potential practical application of this technology in improving the user experience in computer interactions. The implication of this research is a contribution to the development of more intuitive and effective user interfaces through the integration of hand gesture technology in virtual mouse applications.

Keywords: Virtual Mouse, Hand Gesture, Convolutional Neural Network

Abstrak

Pengembangan teknologi interaksi manusia dan komputer terus berkembang, terutama dalam konteks pengenalan gestur tangan untuk aplikasi pengendalian perangkat. Salah satu aplikasi yang menarik adalah perancangan sebuah sistem mouse virtual berbasis pengenalan gestur tangan menggunakan *Convolutional Neural Network* (CNN). Penelitian ini dilakukan untuk mengatasi keterbatasan perangkat input konvensional dan memanfaatkan teknologi modern untuk menciptakan sistem kontrol yang lebih canggih dan adaptif. Penelitian ini fokus pada pengembangan sistem yang mampu mengidentifikasi dan menginterpretasikan gestur tangan secara akurat dalam lingkungan real-time. Sistem yang diusulkan memanfaatkan kamera RGB untuk menangkap citra gestur tangan pengguna, yang kemudian diproses oleh CNN yang telah dilatih sebelumnya untuk mengenali gestur yang berbeda. Eksperimen dilakukan dengan menggunakan dataset gestur yang mencakup variasi pose dan orientasi tangan untuk mengevaluasi kinerja sistem dalam mengenali gestur dengan tingkat akurasi yang memadai. Hasil penelitian menunjukkan bahwa sistem dapat berhasil mengontrol pergerakan kursor pada layar komputer dengan akurasi yang tinggi, memvalidasi potensi aplikasi praktis dari teknologi ini dalam meningkatkan pengalaman pengguna dalam interaksi komputer. Implikasi dari penelitian ini adalah kontribusi terhadap pengembangan antarmuka pengguna yang lebih intuitif dan efektif melalui integrasi teknologi gestur tangan dalam aplikasi mouse virtual.

Kata kunci: Mouse Virtual, Gestur Tangan, Convolutional Neural Network

1. PENDAHULUAN

Dalam era teknologi informasi yang terus berkembang, interaksi antara manusia dan komputer menjadi kunci utama dalam memastikan pengalaman pengguna yang efisien dan intuitif. Salah satu aspek penting dalam hal ini adalah perangkat masukan, khususnya dalam konteks penggunaan *mouse* komputer. Penggunaan *mouse* konvensional telah memberikan kontribusi besar dalam interaksi manusia dan komputer selama beberapa dekade. Namun, ada beberapa permasalahan dan tantangan yang terkait dengan penggunaan *mouse* konvensional yang mungkin timbul dari waktu ke waktu.

Masalah yang dihadapi ketika menggunakan *mouse* konvensional cenderung membatasi fleksibilitas gerakan dan memerlukan ruang fisik yang lebih besar. Batasan-batasan dalam penggunaannya, seperti harus digunakan pada alas yang datar, *mouse* dapat mengalami kendala dalam penggunaannya pada permukaan kaca, *mouse* juga dapat mengalami kendala dalam penggunaannya pada permukaan yang licin atau halus yang dapat membuat sensor *mouse* kesulitan untuk melacak gerakan dengan akurat.

Pengembangan sistem *virtual mouse* dengan menggunakan pengenalan gestur tangan menjadi solusi inovatif untuk meningkatkan fleksibilitas dan kenyamanan interaksi. Dengan gestur tangan, pengguna dapat berinteraksi dengan perangkat tanpa memerlukan permukaan datar, membebaskan pengguna dari keterbatasan *mouse* konvensional. Dengan memanfaatkan gerakan dan posisi tangan, teknologi ini memungkinkan pengguna untuk berkomunikasi dengan komputer

secara langsung tanpa perlu perangkat keras tambahan.

Pada era sekarang, *Convolutional Neural Network* (CNN) telah menjadi pondasi dalam banyak aplikasi pengenalan gambar, termasuk pengenalan pola pada gestur tangan. CNN adalah jenis arsitektur jaringan saraf tiruan (*neural network*) yang secara khusus dirancang untuk mengolah dan menganalisis data citra. CNN memiliki arsitektur *network* yang terdiri dari ratusan *layer*. CNN memproses citra melalui *network layer* dan menghasilkan *output* pada kelas tertentu. Dengan memanfaatkan metode CNN, implementasi pada perancangan sistem *virtual mouse* dapat memberikan respons yang cepat dan akurat terhadap gerakan tangan pengguna.

Pada penelitian ini, akan dilakukan penelitian dan perancangan sistem *virtual mouse* yang menggabungkan konsep gestur tangan dengan CNN. Tujuan utama adalah menciptakan antarmuka yang lebih natural dan efisien bagi pengguna dalam mengontrol *mouse* komputer tanpa perlu perangkat keras tambahan.

2. METODE PENELITIAN

2.1 Rancangan Penelitian, Metode Pengumpulan Data, Teknik Analisis Sistem, Teknik Perancangan Sistem

2.1.1 Rancangan Penelitian

Penulis menggunakan Desain Penelitian Deskriptif dan Eksperimental, penulis melakukan percobaan dan pengujian dengan cara mempelajari literatur-literatur yang berhubungan dengan materi perancangan sistem.

2.1.2 Metode Pengumpulan Data

Penulis mengumpulkan informasi dan data dari buku-buku ilmiah, karya ilmiah, jurnal ilmiah, skripsi, dan sumber-sumber tertulis yang dipublikasikan di berbagai media massa. Informasi dan data dapat berupa teori-teori yang mendasari masalah dan bidang yang akan diteliti oleh penulis.

2.1.3 Teknik Analisis Sistem

Teknik analisis sistem yang digunakan penulis dalam penelitian ini adalah teknik berorientasi objek dengan alat permodelan, yaitu *Unified Modeling Language* (UML) yang bertujuan untuk menggambarkan proses kerja dari perangkat yang ada.

2.1.4 Teknik Perancangan Sistem

Teknik perancangan sistem dalam penelitian ini adalah menggunakan bahasa pemrograman Python sebagai platform untuk membangun sistem.

2.2 Landasan Teori

2.2.1 Pembelajaran Mesin (Machine Learning)

Machine learning merupakan bagian dari ilmu kecerdasan buatan yang menggunakan teknik statistika untuk memproses data. Sistem melakukan pembelajaran terhadap data untuk menghasilkan informasi. *Machine learning* memiliki beragam algoritma dengan model pemrosesan yang bervariasi^[1]. Pembelajaran mesin (*machine learning*) merupakan bidang dalam kecerdasan buatan yang fokus pada pengembangan algoritma dan model komputer yang memungkinkan sistem untuk belajar dari data dan pengalaman

2.2.2 Kecerdasan Buatan

Artificial Intelligence (AI) adalah bidang yang berkembang pesat yang berfokus pada penciptaan sistem komputer yang mampu melakukan tugas-tugas yang membutuhkan kecerdasan buatan^[2]. Kecerdasan buatan merupakan bidang dalam ilmu komputer yang bertujuan untuk menciptakan sistem atau mesin yang memiliki kemampuan untuk melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia, seperti pembelajaran, penciptaan, dan pengenalan gambar.

2.2.3 Convolutional Neural Network (CNN)

Convolutional Neural Network atau *ConvNet* adalah arsitektur pembelajaran mendalam yang dapat belajar otomatis dari data masukan tanpa perlu *feature extractions*. CNN bertugas mereduksi gambar menjadi bentuk yang lebih mudah diproses tanpa menghilangkan fitur penting sehingga hasil prediksi yang didapatkan baik^[3]. *Convolutional Neural Network* (CNN) merupakan salah satu *deep learning* yang mampu melakukan pengenalan gambar dan klasifikasi gambar.

2.2.4 OpenCV (Open Source Vision Library)

OpenCV adalah perpustakaan pengolahan citra dan visi komputer sumber terbuka yang sangat populer dan banyak digunakan. Dirancang untuk membantu pengembangan aplikasi visi komputer, *OpenCV* menyediakan berbagai fungsi dan algoritma seperti analisis citra, deteksi objek, pengenalan pola, pelacakan gerakan, kalibrasi kamera, dan banyak lagi^[4]. *OpenCV* merupakan perpustakaan sumber terbuka yang menyediakan berbagai algoritma dan fungsi untuk pengolahan citra dan penglihatan komputer. *OpenCV* dirancang untuk membantu *programmer* untuk membuat aplikasi yang melibatkan analisis gambar dan pengenalan pola.

2.2.5 Visual Studio Code

Visual Studio Code merupakan *software code editor* yang bersifat *open source* dan dijalankan di berbagai sistem operasi untuk membangun sebuah aplikasi^[5]. *Visual Studio Code* merupakan sebuah *software code editor* yang bersifat *open source* yang dapat dijalankan di berbagai sistem operasi.

2.2.6 Python

Pemrograman bahasa *Python* merupakan bahasa pemrograman yang tidak menggunakan *compiler*. Dengan sifat *open-source* yang dimilikinya, pengguna dapat mempelajari *Python* dengan mudah karena bahasa ini dapat digunakan dalam membuat situs, mengembangkan situs, mengembangkan video *game*, membangun *GUI Desktop*, dan mengembangkan perangkat lunak^[6]. *Python* merupakan bahasa pemrograman yang bersifat *open source* yang digunakan

dalam berbagai bidang seperti pengembangan *web*, pengembangan perangkat lunak, pengembangan *video game*, dan kecerdasan buatan.

2.2.7 Gestur Tangan (Hand Gesture)

Gestur adalah suatu bentuk komunikasi non-verbal dengan aksi tubuh yang terlihat mengomunikasikan pesan-pesan tertentu, baik sebagai pengganti wicara atau bersamaan dan parallel dengan kata-kata. Gestur mengikuti pergerakan dari tangan, wajah, atau bagian lain dari tubuh^[7]. Gestur tangan merupakan gerakan atau pose tangan yang digunakan sebagai bentuk komunikasi atau ekspresi. Dalam konteks teknologi, gestur tangan juga dapat merujuk pada penggunaan gerakan tangan untuk berinteraksi dengan perangkat lunak seperti kontrol gerak (*motion control*) dan perangkat lain yang mendukung interaksi berbasis gestur.

2.2.8 Mouse Komputer

Mouse komputer merupakan perangkat keras yang termasuk dalam golongan perangkat *input* (masukan). Fungsi *mouse* komputer adalah mengatur pergerakan kursor secara cepat, selain itu juga untuk memberikan suatu perintah dengan hanya tombol pada *mouse* komputer^[8]. *Mouse* komputer merupakan perangkat *input* yang digunakan untuk mengendalikan gerakan kursor pada layar komputer.

2.2.9 Pengolahan Citra

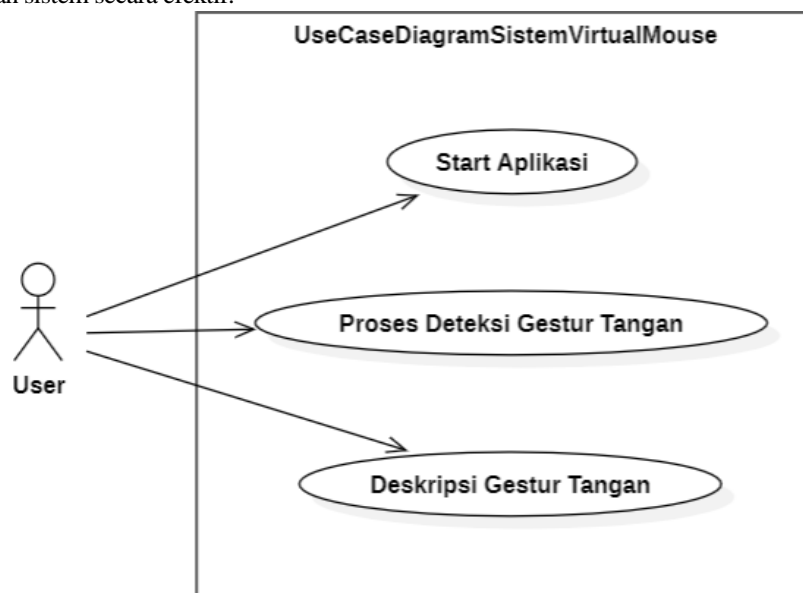
Pengolahan citra adalah suatu implementasi dari proses digital pada data dua dimensi. Dalam sebuah citra yang mengalami penurunan kualitas karena mengandung derau (*noise*), kabur (*blurring*), kurang tajam, kontras, buruk dan sebagainya akan sulit di interpretasi baik oleh manusia maupun sistem^[9]. Pengolahan citra merupakan sebuah proses analisis citra (*image*) untuk meningkatkan kualitas citra, mengekstrak informasi yang berguna, atau mengubah karakteristik citra untuk tujuan tertentu.

2.2.10 Deteksi Objek

Deteksi objek adalah teknologi komputer yang terkait dengan visi komputer dan pemrosesan gambar yang berkaitan dengan mendeteksi instance objek semantic dari kelas tertentu (seperti manusia, bangunan, atau mobil) dalam gambar dan video digital^[10]. Deteksi objek adalah proses dalam pengolahan gambar dan penglihatan computer yang bertujuan untuk mengidentifikasi keberadaan dan lokasi objek tertentu di dalam gambar.

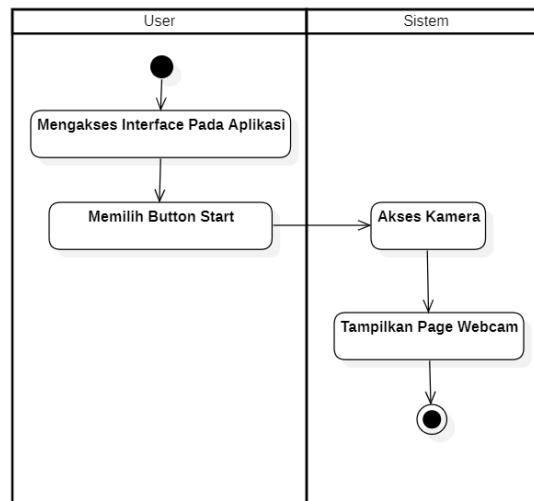
2.2.11 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan pemodelan untuk visualisasi, merancang dan mendokumentasikan sistem informasi ataupun perangkat lunak. *Unified Modeling Language* (UML) lebih simpelnya merupakan istilah sebuah bahasa berbasis objek dan teknik diagram untuk memodelkan suatu proyek pengembangan sistem dari analisis hingga desain^[11]. *Unified Modeling Language* (UML) merupakan bahasa pemodelan visual yang digunakan untuk merancang dan mendokumentasikan sistem perangkat lunak secara terstruktur dan terstandar. UML digunakan secara luas dalam pengembangan perangkat lunak sebagai alat untuk merancang, mengkomunikasikan, dan mendokumentasikan sistem secara efektif.



Gambar 1. Diagram Use Case Sistem Virtual Mouse

Diagram use case yang menggambarkan interaksi antara *user* dan sistem, mencakup aksi seperti memulai kamera, proses deteksi gestur tangan, dan memproses perintah. Sistem pengenalan gestur tangan memiliki *use case* yaitu “*Start Aplikasi*”, “*Proses Deteksi Gestur Tangan*”, dan “*Deskripsi Gestur Tangan*”, sementara aktornya menjadi *user* yang menggunakan sistem ini untuk mengidentifikasi gestur tangan yang kemudian akan diproses sistem dalam proses pengenalan gestur tangan dan menghasilkan sebuah perintah atau aksi.



Gambar 2. Activity Diagram Proses Deteksi Gestur Tangan

Diagram aktivitas pada Gambar 2 memperlihatkan alur kerja proses deteksi gestur tangan dari awal hingga akhir. Ini mencakup aktivitas seperti *start webcam*, mendeteksi gestur tangan, pengenalan gestur tangan, memproses perintah, dan menampilkan hasil dari pemrosesan. Pada saat halaman *Webcam* dimulai, sistem akan mendeteksi gestur tangan pengguna kemudian akan diproses untuk menghasilkan suatu perintah.

2.3 Lingkungan Pengembangan Sistem

Pengembangan sistem menggunakan bahasa pemrograman *Python* sebagai inti pengkodean, dengan *Visual Studio Code* sebagai lingkungan pengembangan untuk pembuatan antarmuka pengguna dan penulisan kode sistem. Arsitektur yang diadopsi dalam pengenalan gambar adalah *Convolutional Neural Network (CNN)*. Pengukuran performa sistem menggunakan metrik akurasi (*accuracy*) dan metrik pengukuran yang melibatkan *Confusion Matrix* untuk mengukur kinerja pengenalan dan klasifikasi gambar. Sistem operasi yang menjadi basis pengembangan sistem ini adalah Windows 11, Windows 11 dipilih karena kompatibilitasnya dengan sebagian besar perangkat keras dan perangkat lunak yang digunakan dalam pengembangan. Komputer yang digunakan untuk pengembangan sistem memiliki spesifikasi yang mendukung kebutuhan komputasi yang diperlukan, termasuk SSD dengan kapasitas 512G, memori RAM sebesar 12GB, prosesor yang digunakan adalah Intel Core i5-12450HX dengan UHD Graphics Family dengan kecepatan 2.4GHz, memberikan performa komputasi yang handal dan responsif.

2.4.2.4 Dataset

Dataset merupakan kumpulan data yang disusun secara terstruktur atau terorganisir dalam bentuk yang dapat diakses dan diolah oleh komputer yang digunakan untuk analisis, penelitian, maupun pengembangan sistem. Kualitas *dataset* sangat mempengaruhi keakuratan dan kehandalan analisis yang dilakukan sistem. Pada implementasi ini menggunakan 6 kelas, yaitu *index*, *v*, *ok*, *four*, *five*, dan *fist*.



Gambar 3. Contoh Gambar untuk Gestur Tangan Index

Gambar 3 merupakan *dataset* yang digunakan untuk jenis gestur tangan *Index* yang terdiri dari 600 gambar dengan dimensi 48x48 piksel. Setiap gambar direpresentasikan dalam format JPG.



Gambar 4. Contoh Gambar untuk Gestur Tangan V

Gambar 4 merupakan *dataset* yang digunakan untuk jenis gestur tangan *V* yang terdiri dari 600 gambar dengan dimensi 48x48 piksel. Setiap gambar direpresentasikan dalam format JPG.



Gambar 5. Contoh Gambar untuk Gestur Tangan Ok

Gambar 5 merupakan *dataset* yang digunakan untuk jenis gestur tangan *Ok* yang terdiri dari 600 gambar dengan dimensi 48x48 piksel. Setiap gambar direpresentasikan dalam format JPG.



Gambar 6. Contoh Gambar untuk Gestur Tangan Four

Gambar 6 merupakan *dataset* yang digunakan untuk jenis gestur tangan *Four* yang terdiri dari 600 gambar dengan dimensi 48x48 piksel. Setiap gambar direpresentasikan dalam format JPG.



Gambar 7. Contoh Gambar untuk Gestur Tangan Five

Gambar 7 merupakan *dataset* yang digunakan untuk jenis gestur tangan *Five* yang terdiri dari 600 gambar dengan dimensi 48x48 piksel. Setiap gambar direpresentasikan dalam format JPG.



Gambar 8. Contoh Gambar untuk Gestur Tangan Fist

Gambar 8 merupakan *dataset* yang digunakan untuk jenis gestur tangan *Fist* yang terdiri dari 600 gambar dengan dimensi 48x48 piksel. Setiap gambar direpresentasikan dalam format JPG.

Dataset ini menjadi representasi visual yang bervariasi dari gestur tangan yang digunakan dalam pengembangan *model* untuk melatih dan menguji algoritma dalam mengenali dan mengklasifikasikan jenis gestur tangan. Dengan jumlah yang cukup dan bervariasi, *dataset* ini mampu meningkatkan keakuratan dan generalisasi *model* yang dikembangkan untuk mengidentifikasi jenis gestur tangan. Kualitas *dataset* sangat penting karena dapat mempengaruhi hasil analisis atau *model* yang dihasilkan. *Dataset* yang buruk atau tidak lengkap dapat mengarah pada kesimpulan yang salah atau *model* yang tidak dapat diandalkan. Oleh karena itu, sebelum melakukan atau membangun *model* menggunakan *dataset*, penting untuk melakukan evaluasi mendalam terhadap kualitas *dataset* tersebut. Langkah-langkah pra-pemrosesan yang tepat dapat membantu memastikan bahwa *dataset* memiliki kualitas yang memadai untuk mendukung hasil yang akurat dan *model* yang dapat diandalkan dalam sistem atau aplikasi.

Tabel 1. Penjelasan Rancangan Gestur Tangan

No	Gestur Tangan	Keterangan
1	<i>Index Gesture</i>	Gestur ini berfungsi sebagai <i>left click</i> untuk menunjuk atau memilih <i>file</i> atau mengklik <i>file</i> tertentu.
2	<i>V Gesture</i>	Gestur ini berfungsi sebagai <i>right click</i> untuk menampilkan menu konteks.
3	<i>Ok Gesture</i>	Gestur ini berfungsi sebagai <i>double click</i> untuk membuka atau mengeksekusi <i>shortcut</i> , folder, <i>file</i> , atau program.
4	<i>Four Gesture</i>	Gestur ini berfungsi untuk menjalankan perintah <i>vertical scroll up</i> .
5	<i>Five Gesture</i>	Gestur ini berfungsi untuk menjalankan perintah <i>vertical scroll down</i> .
6	<i>Fist Gesture</i>	Gestur ini berfungsi untuk mendapatkan posisi kursor dan tidak melakukan aksi.

2.5 Pengembangan Model

Proses pengembangan *model* pada sistem ini menggunakan *TensorFlow-Keras* untuk klasifikasi gambar gestur tangan. Tahapan utamanya terdiri dari pemrosesan *dataset*, pembuatan dan penyetelan *model*, pelatihan, evaluasi *model*, dan penyimpanan *model* yang telah dilatih. Pada tahapan pemrosesan *dataset*, tahap ini membagi *dataset* gambar gestur tangan ke dalam *set* pelatihan, validasi dan *testing* berdasarkan kelasnya dengan perbandingan *split* data 80:10:10. Ukuran gambar *input* ditetapkan menjadi (48, 48, 1). Hal ini menunjukkan bahwa gambar masukan diharapkan memiliki resolusi 48x48 piksel dengan 1 saluran warna (*Grayscale*), sesuai dengan format yang umum digunakan dalam *model Convolutional Neural Network* (CNN). Pada tahapan augmentasi data, penggunaan *ImageDataGenerator* untuk melakukan augmentasi data pada gambar seperti *rescale*, *rotation*, *width shift*, *height shift*, *zoom*, dan *horizontal flip*.

Pada tahapan *preprocessing*, penggunaan *rescale* = 1.0 / 255 untuk normalisasi intensitas piksel citra ke rentang 0-1. Ini adalah langkah umum dalam pra-pemrosesan gambar yang membantu dalam konvergensi pelatihan *model*. Pada tahapan klasifikasi kelas, terdapat pengaturan *class_mode* = 'categorical' di *generator data*, menandakan bahwa *model* melakukan klasifikasi dalam beberapa kelas. Jumlah kelas yang ada pada sistem adalah 6, sesuai dengan *layer output* yang memiliki 6 *neuron* dengan aktivasi *softmax*. Setiap *neuron* pada *layer* ini mewakili salah satu kelas.

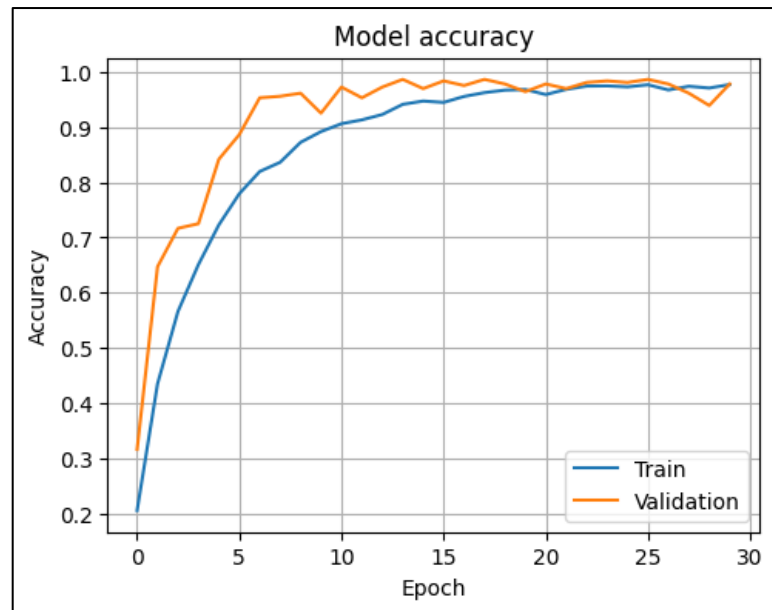
Pada tahapan pelatihan *model*, *model* tersebut dilatih menggunakan *dataset* pelatihan dengan konfigurasi *hyperparameter* terbaik. Setelah *model* dilatih, tahap evaluasi dilakukan menggunakan *dataset* validasi. Evaluasi ini mencakup perhitungan metrik akurasi (*accuracy*) dan *Confusion Matrix* untuk memahami performa *model* dalam klasifikasi berbagai kelas gestur tangan. Pada tahapan terakhir, *model* yang telah dilatih dan dievaluasi disimpan dalam format *KerasModel*. Ini memungkinkan penggunaan *model* di masa mendatang untuk melakukan klasifikasi gambar gestur tangan ke dalam kelas yang sesuai. Keseluruhan proses ini sangat bergantung pada struktur *dataset* yang diberikan dan menggunakan teknik augmentasi gambar serta penyetelan *parameter* untuk meningkatkan performa *model*.

3. HASIL DAN PEMBAHASAN

Grafik *training* dan *validation accuracy* menggambarkan evolusi performa *model* selama proses pelatihan. Pada awalnya, saat *model* mulai dilatih, kedua metrik cenderung rendah dan terpisah. Namun, seiring berjalannya *epoch*, terlihat peningkatan yang konsisten dalam kedua metrik tersebut. Ini mengindikasikan bahwa *model* sedang belajar dari data latihnya. Akhirnya, terlihat hasil yang cukup stabil dimana kedua akurasi baik pada data latih maupun validasi, semakin mendekati nilai tertinggi. Hal ini menunjukkan bahwa *model* telah belajar dengan baik dan mampu memberikan prediksi yang kuat tidak hanya pada data latih tetapi juga pada data yang belum pernah dilihat sebelumnya (data validasi).

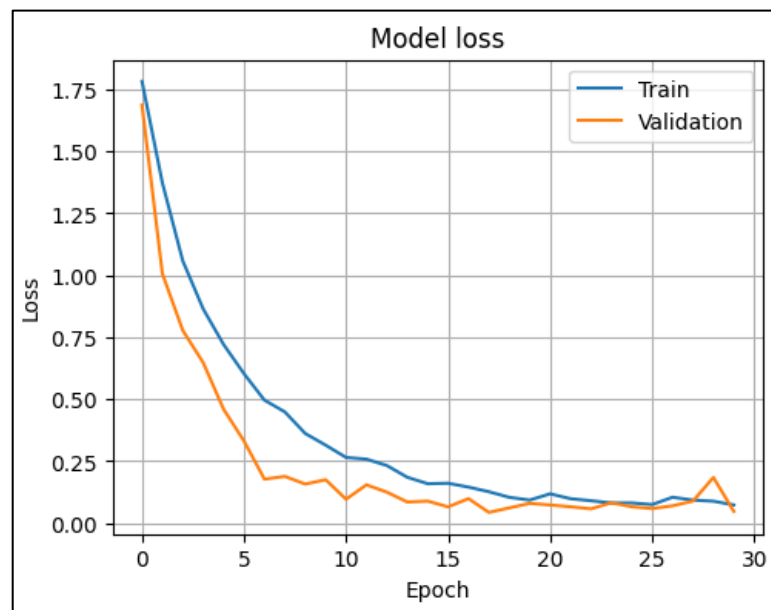
Sementara kedua akurasi tersebut mendekati nilai maksimum, yang perlu diperhatikan adalah seberapa jauh kedua garis ini saling berdekatan. Semakin dekat kedua garis, semakin stabil dan konsisten performa *model* pada kedua set data tersebut, menunjukkan kekuatan dan keandalannya dalam menggeneralisasi pola dari data latih ke data validasi. Dalam grafik tersebut, konvergensi yang baik dan perbedaan yang minimal antara kedua metrik tersebut menunjukkan bahwa *model* memiliki kemampuan yang kuat untuk menggeneralisasi pada data baru, menggambarkan konvergensi dan keandalan performanya yang tinggi dan kecocokan *model* terhadap data.

Grafik *training loss* dan *validation loss* membawa informasi penting tentang bagaimana *model* kita belajar dari data yang diberikan selama proses pelatihan. Ketika proses pelatihan dimulai, kedua *loss* cenderung tinggi dan terpisah jauh, menandakan ketidakcocokan *model* terhadap data. Namun, seiring berjalannya waktu (*epoch*), *loss* mulai menurun secara bertahap untuk kedua set data. Perlahan-lahan, terjadi penyatuan antara kedua *loss* tersebut. Penurunan yang stabil dalam *loss* validasi menunjukkan bahwa *model* tidak hanya belajar dari data latih tetapi juga mampu memberikan prediksi yang akurat pada data yang belum pernah dilihat sebelumnya (data validasi). Ketika kedua *loss* ini semakin mendekati nilai minimum-nya, maka yang perlu diperhatikan adalah seberapa dekat kedua garis tersebut. Jika kedua garis *loss* tersebut berada dalam jarak yang minimal, itu menandakan bahwa *model* cenderung memiliki kemampuan yang baik untuk menggeneralisasi dari data latih ke data validasi. Ini menunjukkan keandalan *model* dalam memprediksi data baru.



Gambar 9. Evaluasi Training dan Validation Accuracy

Kesamaan antara kedua *loss* ini mencerminkan tingkat yang baik dari generalisasi *model* terhadap *dataset* baru, menunjukkan stabilitas dan konsistensinya dalam memberikan prediksi yang kuat. Semakin kecil perbedaan antara kedua *loss* tersebut menandakan bahwa *model* mampu menangkap pola yang lebih kompleks dalam *dataset* tanpa *overfitting* atau *underfitting* yang signifikan. Hal ini menandakan bahwa kinerja *model* dalam hal akurasi prediksi tidak bervariasi secara signifikan ketika diukur dengan cara yang berbeda. Ketika *model* dapat menggeneralisasi dengan baik terhadap *dataset* baru, artinya *model* tidak hanya menghafal data pelatihan tetapi juga mampu menangkap pola umum yang ada dalam data, menandakan bahwa *model* memiliki kemampuan untuk mempelajari fitur-fitur yang relevan dan mengabaikan *noise* atau variabilitas yang tidak signifikan dalam data.

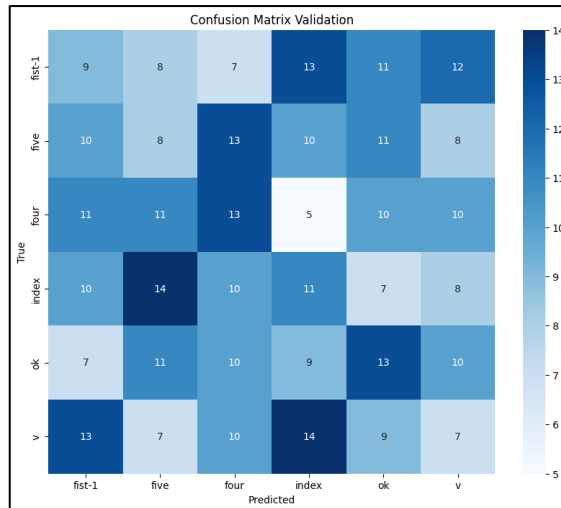


Gambar 10. Evaluasi Training dan Validation Loss

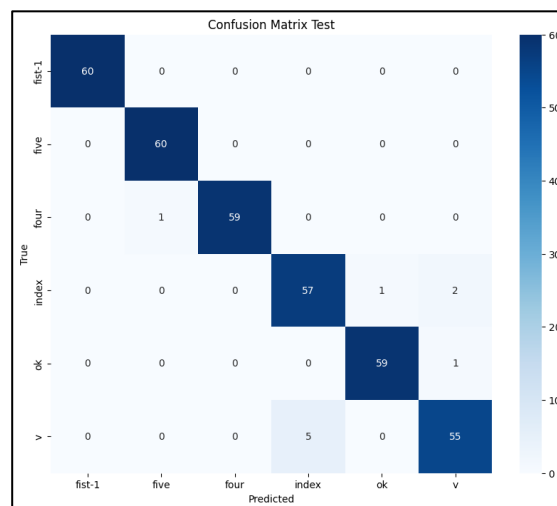
Confusion matrix pada tahap validasi yang belum menunjukkan pola diagonal secara konsisten mungkin menandakan adanya kompleksitas dalam pengklasifikasian data. Hal ini bisa terjadi karena sejumlah faktor, seperti kurangnya representasi kelas tertentu dalam *dataset* validasi atau ketidaktertangkapan variasi yang cukup dari pola yang kompleks dari *model*.

Namun, pada tahap pengujian (*testing*), menemukan pola diagonal yang konsisten dalam *confusion matrix* menunjukkan bahwa *model* berhasil mengklasifikasikan data dengan tingkat keakuratan yang lebih baik. Hasil ini mengindikasikan bahwa *model* cenderung memiliki kemampuan yang baik dalam memprediksi kelas data yang baru, dengan lebih sedikit kesalahan klasifikasi dibandingkan dengan tahap validasi. Meskipun begitu, penting untuk tetap

melakukan evaluasi terhadap kinerja *model* untuk memastikan *model* bekerja dengan baik.



Gambar 11. Confusion Matrix untuk Validation Data



Gambar 12. Confusion Matrix untuk Test Data

Sistem klasifikasi jenis gestur tangan berhasil mencapai akurasi sebesar 98.66 persen, *precision* sebesar 97.22 persen, *recall* sebesar 97.22 persen, dan *F1-score* sebesar 97.22 persen yang menunjukkan bahwa sistem dapat mengklasifikasi jenis gestur tangan dari gambar yang belum pernah dipelajari oleh sistem dengan sangat baik.

Selama 30 *epoch* pelatihan *model*, terjadi peningkatan atau penurunan akurasi dan *loss* ketika melakukan proses pelatihan. Mulai dari akurasi sekitar 18.73 persen pada awal proses, *model* secara konsisten meningkat ke tingkat akurasi hampir sempurna sekitar 97.18 persen saat mendekati akhir periode. Evaluasi pada data validasi menunjukkan kenaikan yang signifikan dari 31.67 persen menjadi 97.78 persen, menggambarkan kemampuan *model* dalam membedakan jenis-jenis gestur tangan dengan baik. Namun, fluktuasi terjadi pada akurasi validasi di beberapa titik, terutama pada *epoch* 3 hingga 7, menandakan kesulitan dalam pengenalan jenis gestur tangan pada sebagian data validasi. Meski demikian, akurasi yang tinggi pada data validasi menunjukkan potensi *model* dalam mengenali jenis gestur tangan secara efektif.

Tabel 2. Hasil Training Model Untuk Testing Data

Validation Accuracy	Loss	Accuracy	Precision	Recall	F1-score
97.22%	5.80%	98.66%	97.24%	97.22%	97.22%

Pada awal pelatihan, *model* menunjukkan peningkatan yang stabil dalam akurasi. Dalam lima *epoch* pertama, terlihat bahwa *model* mulai memahami pola-pola yang ada dalam data dengan baik, dengan ditandai dengan kenaikan yang konsisten dalam akurasi pada data validasi. Terjadi fluktuasi yang mencolok pada pertengahan pelatihan, khususnya pada *epoch* ke-10. Terjadinya penurunan yang signifikan dalam akurasi pada data validasi, ini mengindikasikan adanya

kemungkinan *overfitting* atau ketidakcocokan parameter pada tahap tersebut.

Meskipun mengalami fluktuasi, *model* berhasil pulih dengan baik pada *epoch* ke-13, menunjukkan peningkatan yang dramatis dalam akurasi pada data validasi. Ini menandakan bahwa *model* mampu menyesuaikan diri kembali dan mempelajari pola yang rumit dari data. Setelah pemulihan tersebut, *model* menunjukkan konsistensi yang luar biasa dalam memprediksi dengan akurat, dengan tingkat akurasi pada data validasi yang sangat tinggi pada *epoch* ke-26 dan ke-30. Hal ini menunjukkan keandalan *model* dalam pemrosesan data yang kompleks. Secara keseluruhan, meskipun terjadi fluktuasi pada beberapa titik, kemampuan *model* untuk pulih dan mencapai performa yang sangat baik menunjukkan kemampuannya yang sangat baik dalam mempelajari pola-pola yang rumit dalam data dengan tingkat akurasi yang tinggi pada data yang belum dipelajari.

Tabel 3. Hasil Training Model

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy
1	1.7918	18.73%	1.6870	31.67%
2	1.5108	38.79%	1.0064	64.72%
3	1.1284	53.24%	0.7774	71.67%
4	0.8886	63.88%	0.6475	72.50%
5	0.7707	69.43%	0.4595	84.17%
6	0.6238	77.31%	0.3315	88.61%
7	0.4968	81.28%	0.1778	95.28%
8	0.4722	82.30%	0.1888	95.56%
9	0.3838	85.82%	0.1579	96.11%
10	0.3453	87.06%	0.1753	92.50%
11	0.2568	91.19%	0.0968	97.22%
12	0.2591	90.76%	0.1548	95.28%
13	0.2462	91.51%	0.1253	97.22%
14	0.1776	94.36%	0.0856	98.61%
15	0.1721	94.18%	0.0891	96.94%
16	0.1682	93.81%	0.0661	98.33%
17	0.1510	95.57%	0.0992	97.50%
18	0.1279	96.14%	0.0435	98.61%
19	0.1028	96.60%	0.0617	97.78%
20	0.0964	96.05%	0.0801	96.39%
21	0.1303	95.50%	0.0735	97.78%
22	0.1062	96.63%	0.0664	96.94%
23	0.0907	97.56%	0.0579	98.06%
24	0.0955	96.79%	0.0823	98.33%
25	0.0868	97.06%	0.0663	98.06%
26	0.0805	97.28%	0.0589	98.61%
27	0.1070	96.65%	0.0702	97.78%
28	0.0852	97.48%	0.0893	96.11%
29	0.1014	96.58%	0.1846	93.89%
30	0.0837	97.18%	0.0476	97.78%

4. KESIMPULAN

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan dalam skripsi ini, dapat diambil beberapa kesimpulan sebagai berikut:

- Sistem virtual mouse yang dirancang menggunakan teknik *hand gesture recognition* berbasis *Convolutional Neural Network* (CNN) dapat diimplementasikan dengan baik. Sistem ini mampu mengenali berbagai gerakan tangan seperti perintah mouse, seperti klik kiri, klik kanan, scroll, dan gerakan kursor.
- Akurasi pengenalan gerakan tangan pada sistem ini mencapai tingkat yang memadai, dengan hasil pengujian menunjukkan tingkat kesalahan rendah. Ini menandakan bahwa CNN efektif dalam mengidentifikasi berbagai gerakan tangan dan menerjemahkannya menjadi perintah mouse yang sesuai.
- Sistem menunjukkan kinerja yang stabil dalam kondisi pengujian yang berbeda, dengan respons yang cukup cepat dan konsisten dalam menerjemahkan Gerakan tangan menjadi perintah mouse.
- Kelebihan dari sistem ini termasuk kemampuannya dalam memberikan alternatif kontrol komputer tanpa memerlukan perangkat keras tambahan selain kamera. Namun, keterbatasan sistem ini mencakup ketergantungan pada pencahayaan lingkungan dan kebutuhan untuk pelatihan model yang lebih lanjut untuk meningkatkan akurasi pada berbagai kondisi tangan dan gerakan.

5. SARAN

Untuk pengembangan lebih lanjut dan peningkatan sistem, penulis memberikan beberapa saran sebagai berikut:

- a) Peningkatan Model: Melakukan pelatihan lebih lanjut pada model CNN dengan dataset yang lebih besar dan beragam untuk meningkatkan akurasi pengenalan gerakan tangan dalam berbagai kondisi pencahayaan dan variasi gerakan.
- b) Integrasi Fitur Tambahan: Menambahkan fitur tambahan, seperti pengenalan gerakan tangan yang lebih kompleks atau variasi perintah, untuk meningkatkan fungsionalitas sistem dan memperluas penggunaannya.
- c) Optimasi Kinerja: Mengoptimalkan algoritma dan kode untuk mempercepat proses pengenalan dan respons sistem, serta mengurangi latensi yang mungkin terjadi.
- d) Uji Coba di Berbagai Kondisi: Melakukan uji coba sistem di berbagai kondisi lingkungan dan dengan berbagai tipe kamera untuk memastikan sistem dapat berfungsi secara optimal dalam situasi nyata.

UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih yang sebesar-besarnya kepada civitas akademika Fakultas Teknologi Informasi Universitas Widya Dharma Pontianak dan pihak-pihak lain yang memberikan dukungan dalam menyelesaikan penelitian ini dengan baik. Terima kasih.

DAFTAR PUSTAKA

- [1] Setiawan, Wahyudi. (2020). Deep Learning Menggunakan Convolutional Neural Network: Teori dan Aplikasi. Media Nusa Creative.
- [2] Manalu, Sovantro Derisjon, Adinda Naya Laka Chilfi, Ega Arif Surya Baskara, Liyanti, Indah Puspika Widarma, Celena Helga Izdihar, Daniel Siburian, Muhammad Ferlian Adhitama, Mika Apriela Br Situmorang, Tamaro Simamora, Risydi Liana Balqis, Azizah, Habib Dermawan, Muhammad Alief Nurrahman, Muhammad Rifki Januarta, Monasari, Dzihnul Abdi Wicaksono, Rikhesya Anjani Gumanti, Fanny Tamba, Indah Siburian, Gustina Triayu Hana Nazwa, Tiur Pane. (2020). AI: Revolusi Pembelajaran Menerobos Batasan melalui Pemanfaatan Kecerdasan Buatan dalam Pendidikan. CV Brimedia Global.
- [3] Kahlil, Muhammad Rizky Munggaran, Laksono Kurnianguro, Adhiguna Mahendra, Nona Zarima, Fina Noviantika, Alifya Febriana. (2023). Computer Vision Berbasis Deep Learning Untuk Aplikasi Pertanian: Teori dan Praktek. Syiah Kuala University Press.
- [4] Yuadi, Imam. (2023). Forensik Digital dan Analisis Citra. CV. AE Media Grafika.
- [5] Awalina, Eriskiannisa Febrianty Luchia, Woro Isti Rahayu dan Cahyo Prianto. (2023). Penerapan Metode Least Cost Untuk Menentukan Biaya Minimum dan Jumlah Barang yang Dikirim Pada Aplikasi Transport Management System. PT. Penerbit Buku Pedia.
- [6] Setiawan, Gabriella Alicia dan Evelyn Vania. (2022). Praktek Pemrograman C++ dan Python. SCU Knowledge Media.
- [7] Haryanto dan Helena Ras Ulina Sembiring. (2021). Membangun Pribadi Prima Dalam Pelayanan Publik. Media Nusa Creative (MNC Publishing).
- [8] Sumarno. (2020). Pengantar Teknologi Informasi dan Komunikasi. Uwais Inspirasi Indonesia.
- [9] Zonyfar, Candra. (2020). Pengolahan Citra Digital. Desanta Muliavisitama.
- [10] Putro, Eko Cahyono, Rolly Maulana Awangga dan Roni Andarsyah. (2020). Tutorial Object Detection People With Faster Region-Based Convolutional Neural Network (Faster R-CNN). Kreatif Industri Nusantara.
- [11] Zein, Afrizal, Dahlan Susilo, Mustakim, Ryan Effendi, Winny Purbaratri, Achmad Ridwan, Subhan Nooriansyah, Faridatun Nadziroh, Anyan, Ali Ibrahim. (2023). Konsep Dasar Rekayasa Perangkat Lunak. Yayasan Cendikia Mulia Mandiri.