

Implementasi *Virtual Mouse* Berbasis *Hand Landmark Detection* Menggunakan YOLO Untuk Navigasi Tanpa Sentuh

Dana Christin¹, Neny Rosmawarni², Muhammad Panji Muslim³,

S1 Informatika / Fakultas Ilmu Komputer

Universitas Pembangunan Nasional “Veteran” Jakarta

Jalan RS. Fatmawati Raya, Pondok Labu, Cilandak, Jakarta Selatan, 12450

2210511115@mahasiswa.upnvj.ac.id¹, nenyrosmawarni@upnvj.ac.id², muhammadpanji@upnvj.ac.id³

Abstrak. Interaksi manusia dengan komputer yang terus berkembang mendorong kebutuhan akan antarmuka yang lebih intuitif dan minim ketergantungan perangkat keras fisik. *Mouse* konvensional dan *touchpad* memiliki keterbatasan seperti keterbatasan ruang fisik, ketergantungan perangkat keras, serta risiko gangguan *musculoskeletal* akibat penggunaan intensif. Penelitian ini mengimplementasikan sistem *Virtual Mouse* berbasis *Hand Landmark Detection* menggunakan model YOLO11 sebagai solusi navigasi tanpa sentuh yang mendukung sembilan aksi mouse melalui gestur tangan, meliputi *Move Cursor*, *Left Click*, *Right Click*, *Double Click*, *Drag*, *Scroll Up*, *Scroll Down*, *Zoom In*, dan *Zoom Out*. Hasil pengujian menunjukkan tingkat keberhasilan fungsional 99% dari 200 *test case*, latensi rata-rata 54,66 ms pada GPU dan 239,77 ms pada CPU, ketahanan baik pada berbagai kondisi pencahayaan dan latar belakang kompleks, serta penilaian pengguna rata-rata 3,99 dari skala 5,00 dengan seluruh 10 penguji menyatakan sistem layak sebagai alternatif mouse fisik.

Kata Kunci: *Virtual Mouse*, *Hand Landmark Detection*, YOLO11, Gestur Tangan

1 Pendahuluan

Perkembangan teknologi digital telah merevolusi cara manusia bekerja dan berinteraksi [1]. Salah satu bidang yang berkembang pesat adalah *Human-Computer Interaction* (HCI), yang berfokus pada desain interaksi antara manusia dan komputer [2], dengan mouse sebagai salah satu perangkat input utamanya. Namun, *mouse* konvensional memiliki keterbatasan, terutama pada permukaan kaca atau licin di mana sensor optiknya sulit mendeteksi gerakan secara akurat [3]. Penulis sendiri mengalami berbagai kendala serupa, seperti *mouse wireless* yang cepat rusak, *touchpad* yang menyebabkan pegal pada pergelangan tangan, hingga mouse berkabel yang membatasi ruang gerak. Kondisi ini sejalan dengan temuan pada [4] bahwa penggunaan *mouse* secara terus-menerus dapat menyebabkan *musculoskeletal disorders* (MSDs), termasuk risiko *Carpal Tunnel Syndrome* (CTS) akibat posisi pergelangan tangan yang buruk.

Sebagai solusi, sistem *Virtual Mouse* berbasis gestur tangan dikembangkan untuk memungkinkan kontrol kursor tanpa kontak fisik yang lebih ergonomis. Urgensi ini juga didukung survei terhadap 62 responden (88,7% mahasiswa), yang menunjukkan bahwa 96,7% tertarik pada sistem *Virtual Mouse* berbasis kamera sebagai solusi navigasi tanpa sentuh. Meski demikian, penelitian terdahulu masih menyisakan tantangan keandalan (*robustness*) dan kinerja (*performance*). Penelitian pada [3] menemukan model berbasis CNN sangat bergantung pada kondisi pencahayaan. Penelitian pada [5] melaporkan bahwa sistem berbasis Mediapipe dan Faster R-CNN hanya mencapai 7 FPS, yang dinilai "belum baik" untuk interaksi *real-time*. Sebaliknya, arsitektur *single-stage detector* seperti YOLO terbukti lebih unggul; penelitian pada [6] menunjukkan bahwa YOLOv8 nano memiliki latensi lebih rendah, yaitu 17ms berbanding 40ms, serta akurasi lebih tinggi, yaitu 98,54% berbanding 98,26% dibanding Mediapipe. Namun, penelitian pada [7] mencatat bahwa YOLOv5 masih kesulitan mendeteksi tangan dalam pose tak umum, kemungkinan akibat dataset pelatihan yang terbatas, yaitu hanya 216 foto.

Berdasarkan hal tersebut, penelitian ini mengusulkan implementasi *Virtual Mouse* menggunakan YOLO11, yang menawarkan keseimbangan optimal antara akurasi, efisiensi, dan ukuran model untuk aplikasi *real-time*. YOLO12 tidak dipilih karena arsitekturnya yang lebih kompleks justru menghasilkan performa kurang stabil, sebagaimana ditemukan pada [8]. Dengan YOLO11, penelitian ini bertujuan menjawab tantangan keandalan pada berbagai

kondisi pencahayaan dan latar belakang, sekaligus mengatasi kelemahan yang diidentifikasi pada [3].

2 Tinjauan Pustaka

Human-Computer Interaction (HCI) merupakan disiplin yang mempelajari interaksi antara manusia dan teknologi agar berjalan efektif, efisien, dan menyenangkan bagi pengguna [9]. HCI kini bergeser dari interaksi berbasis perangkat fisik seperti *mouse* dan *keyboard* menuju interaksi alami berbasis gestur, seiring berkembangnya teknologi tanpa sentuh (*touchless*) yang memungkinkan pengendalian sistem tanpa kontak fisik langsung [10]. Salah satu bentuk penerapannya adalah *Virtual Mouse*, yaitu inovasi HCI yang memanfaatkan kamera dan algoritma pengolahan citra untuk mendeteksi gerakan tangan dan menerjemahkannya menjadi aksi seperti klik, geser, atau gulir [11]. Teknologi ini bertumpu pada *computer vision* untuk memperoleh informasi relevan dari gambar melalui tugas seperti *object detection* [12][13], serta *landmark detection* menggunakan framework MediaPipe untuk mengidentifikasi titik-titik penting pada objek [14], yang menjadi dasar bagi *hand gesture recognition* dalam menginterpretasikan gerakan tangan sebagai perintah *input* [15]. Kemampuan tersebut didukung oleh *deep learning*, yaitu cabang *machine learning* yang menggunakan jaringan saraf tiruan berlapis untuk mengenali pola kompleks secara otomatis tanpa memerlukan rekayasa fitur manual, dan banyak digunakan dalam pengenalan citra serta deteksi objek [16].

Dari sisi metodologi, perancangan sistem dalam penelitian ini mengacu pada UML (*Unified Modelling Language*), yaitu bahasa pemodelan visual standar yang digunakan untuk menspesifikasikan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. Pemodelan ini dituangkan melalui *Use Case Diagram* untuk menggambarkan fungsionalitas sistem dari sudut pandang pengguna, *Class Diagram* untuk memetakan struktur statis dan hubungan antar komponen sistem, *Activity Diagram* untuk memvisualisasikan alur proses mulai dari inisialisasi kamera hingga eksekusi perintah, serta *Sequence Diagram* untuk menggambarkan urutan interaksi antar objek dalam sistem. Selain itu, pengembangan sistem menggunakan metode *Rapid Application Development* (RAD), yaitu pendekatan yang berfokus pada percepatan proses pembuatan sistem melalui iterasi singkat, keterlibatan pengguna secara aktif, dan penggunaan kembali komponen perangkat lunak, yang terdiri atas empat fase utama yaitu *Requirement Planning*, *User Design*, *Construction*, dan *Cutover* [18]. Pendekatan ini dinilai efektif untuk proyek dengan waktu pengembangan terbatas karena memungkinkan penyempurnaan berkelanjutan melalui kolaborasi antara pengguna dan pengembang.

Dalam pengembangan sistem *Virtual Mouse*, beberapa penelitian terdahulu telah dilakukan dengan berbagai pendekatan metode, namun masing-masing menyisakan celah yang menjadi landasan pengembangan penelitian ini. Ken dkk. mengimplementasikan *Convolutional Neural Network* (CNN) untuk mengenali enam kelas gestur tangan dari citra kamera RGB secara *real-time*, dan berhasil mencapai akurasi testing yang sangat tinggi yaitu 98,66%. Meskipun demikian, sistem ini memiliki keterbatasan besar berupa ketergantungan yang signifikan terhadap kondisi pencahayaan lingkungan, sehingga rentan mengalami kegagalan deteksi saat diterapkan pada kondisi dunia nyata dengan pencahayaan yang dinamis [3]. Pendekatan berbeda dilakukan oleh Hernando, yang menggabungkan MediaPipe untuk pelacakan landmark tangan dengan Faster R-CNN untuk deteksi gestur berdasarkan dataset 900 citra gestur jari tangan. Sistem ini mencapai akurasi ideal 73,16% pada jarak 50 cm, namun kinerja *real-time* yang dihasilkan sangat lambat dengan rata-rata hanya 7 FPS, yang dinilai sendiri oleh peneliti sebagai belum memadai untuk interaksi *real-time* [5]. Krishnamoorthi dkk. mengimplementasikan algoritma YOLOv5 dengan dataset 216 gambar tangan yang dinotasikan ke dalam lima kelas gestur. Hasil penelitian membuktikan bahwa YOLOv5 mampu mendeteksi gerakan tangan secara *real-time* dengan akurasi tinggi dan kecepatan hingga 45 FPS, namun model masih kesulitan mendeteksi tangan dalam pose dan orientasi yang tidak umum, yang kemungkinan disebabkan oleh keterbatasan variasi dataset yang digunakan [7]. Adapun Karthick dkk. mengembangkan model *Enhanced Hand Tracking* (EHT) berbasis YOLOv8 nano yang dilatih dari dataset 1574 gambar tangan dengan empat kategori gestur, serta diintegrasikan dengan PyAutoGUI untuk mengubah koordinat tangan menjadi kontrol *mouse* secara *real-time*. Hasil penelitian menunjukkan bahwa model berbasis YOLOv8 nano secara signifikan mengungguli MediaPipe, baik dari segi akurasi yaitu 98,54% berbanding 98,26%, maupun latensi yaitu 17ms berbanding 40ms [6].

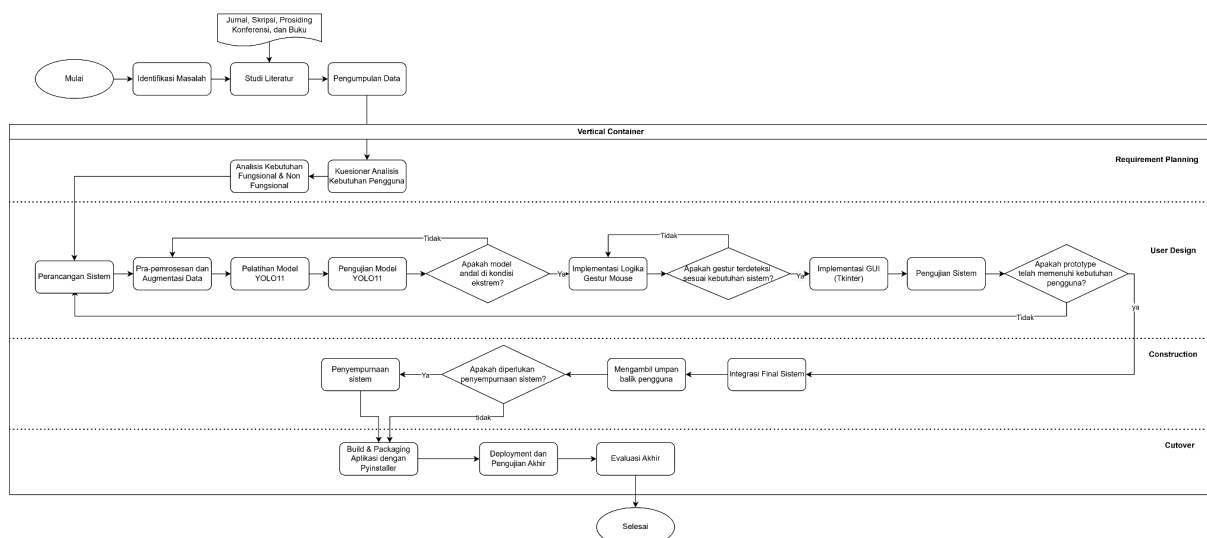
Berdasarkan analisis terhadap penelitian-penelitian tersebut, dapat diidentifikasi bahwa pendekatan berbasis CNN meskipun mampu mencapai akurasi pengujian yang sangat tinggi, memiliki kelemahan fundamental berupa

ketergantungan signifikan terhadap kondisi pencahayaan lingkungan sehingga kurang andal pada kondisi dunia nyata [3]. Di sisi lain, implementasi arsitektur *multi-stage* seperti Faster R-CNN yang digabungkan dengan Mediapipe terbukti tidak efisien untuk aplikasi yang menuntut responsivitas tinggi [5], sementara arsitektur single-stage seperti YOLOv5 masih terkendala keterbatasan variasi dataset dalam mendeteksi pose tangan yang tidak umum [7]. Sebaliknya, penelitian yang membandingkan YOLOv8 nano dengan Mediapipe telah membuktikan bahwa arsitektur YOLO memiliki kinerja yang jauh lebih unggul baik dari sisi akurasi maupun latensi [6]. Temuan inilah yang menjadi dasar kuat untuk menerapkan pembaruan arsitektur ke versi yang lebih baru dalam penelitian ini.

Oleh karena itu, penelitian ini memilih untuk mengimplementasikan arsitektur YOLO11, yang dikembangkan oleh Ultralytics pada tahun 2024 sebagai penerus dari YOLOv8. YOLO11 dirancang untuk melampaui efisiensi dan akurasi model pendahulunya melalui pembaruan yang meningkatkan kecepatan inferensi dan efisiensi pelatihan tanpa mengorbankan presisi deteksi. Salah satu inovasi utamanya adalah integrasi modul C2PSA (*Cross-Stage Partial with Self-Attention*), yang menggabungkan efisiensi komputasi dari arsitektur CSP dengan mekanisme *self-attention* untuk meningkatkan kemampuan model dalam menangkap fitur spasial dan kontekstual yang kompleks, sehingga sensitivitas deteksi menjadi lebih baik pada kondisi menantang seperti oklusi atau latar belakang yang kompleks [16][17]. Pemilihan YOLO11 dalam penelitian ini didasarkan pada keseimbangan optimal yang ditawarkannya antara akurasi, efisiensi, dan ukuran model, sehingga sangat sesuai untuk aplikasi *real-time* seperti *virtual mouse* yang menuntut respons cepat tanpa mengorbankan ketepatan deteksi. Sementara itu, arsitektur yang lebih baru seperti YOLO12 tidak dipilih karena, meskipun memiliki desain yang lebih kompleks, performanya dinilai kurang stabil sehingga kurang ideal untuk pengenalan gestur tangan yang membutuhkan respons cepat dan akurat [8]. Dengan mengimplementasikan YOLO11, penelitian ini bertujuan menjawab tantangan keandalan pada berbagai kondisi pencahayaan dan latar belakang, sekaligus mengatasi kelemahan-kelemahan yang telah diidentifikasi pada penelitian-penelitian terdahulu.

3 Metode

Metode yang digunakan pada penelitian ini adalah *Rapid Application Development* (RAD), yang dipilih karena memungkinkan iterasi cepat dalam perancangan *Virtual Mouse* sehingga proses pengujian dan penyempurnaan fitur dapat dilakukan secara efektif dan efisien. Alur penelitian ditunjukkan pada Gambar 3.1.



Gambar. 1. Tahapan Penelitian

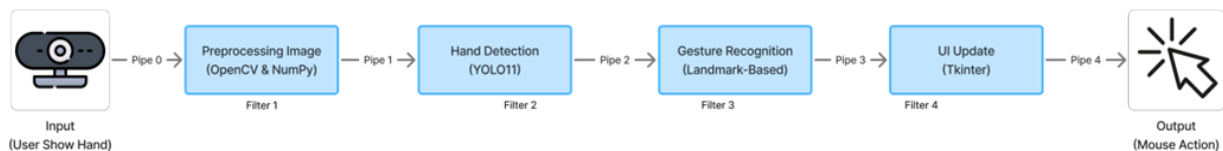
Penelitian ini diawali dengan tahap pra-pengembangan yang mencakup identifikasi masalah, studi literatur, dan pengumpulan data. Permasalahan utama yang melatarbelakangi penelitian ini adalah keterbatasan *mouse* konvensional dan *touchpad*, seperti ketergantungan pada permukaan datar, keterbatasan mobilitas akibat kabel,

serta kerentanan kerusakan, yang diperkuat oleh hasil kuesioner terhadap 62 responden dan pengalaman langsung peneliti. Studi literatur kemudian dilakukan untuk mengkaji konsep *hand gesture recognition* dan model YOLO11, sekaligus mengidentifikasi keterbatasan penelitian terdahulu sebagai dasar penentuan celah penelitian.

Pada tahap pengumpulan data, dataset dikumpulkan secara mandiri dari empat subjek berbeda, menghasilkan total 9.313 gambar setelah proses *cleaning* yang terbagi ke dalam empat kategori, yaitu gestur *mouse* tangan kanan, gestur *mouse* tangan kiri, gestur umum non-*mouse* sebagai data negatif, dan data tanpa tangan. Dataset kemudian dianotasi menggunakan format YOLO11 *pose estimation* dengan *bounding box* dan 21 titik *keypoint*. Selanjutnya, dataset melalui tahap pra-pemrosesan berupa *cleaning*, normalisasi koordinat, dan augmentasi data (Random HSV, *rotation*, dan *copy-paste*, masing-masing 20% dari total data), sebelum akhirnya dibagi dengan rasio 80:20 secara *stratified* menjadi data latih dan data validasi.

Selanjutnya, kebutuhan sistem dianalisis dari dua aspek. Kebutuhan fungsional mencakup fungsi-fungsi utama sistem, mulai dari inialisasi dan pengelolaan antarmuka, deteksi tangan dan identifikasi gestur, interpretasi gestur menjadi aksi mouse, pemrosesan *hand tracking* secara paralel, eksekusi aksi mouse, hingga visualisasi status pada layar, yang dirancang secara modular berbasis *Object-Oriented Programming* (OOP) agar setiap fungsi dapat dikembangkan dan diuji secara independen. Kebutuhan non-fungsional meliputi *responsiveness* dengan target respons di bawah 100 ms, *accuracy* dengan akurasi minimal 90% pada jarak 50–70 cm, serta *usability*, *scalability*, dan *maintainability* untuk memastikan sistem mudah digunakan, dikembangkan, dan dipelihara.

Berdasarkan kebutuhan tersebut, alur kerja sistem dirancang mulai dari akuisisi video, deteksi tangan, ekstraksi *bounding box*, deteksi *hand landmark* melalui YOLO11, interpretasi gestur, hingga eksekusi aksi *mouse* dan umpan balik visual kepada pengguna, mencakup sembilan aksi *mouse* yang dapat dikenali sistem. Arsitektur sistem yang digunakan mengikuti pola *Pipes and Filters*, sebagaimana ditunjukkan pada Gambar 3.2, yang terdiri atas empat *filter* berurutan, yaitu *preprocessing image*, *hand detection* menggunakan YOLO11, *gesture recognition* berbasis *landmark*, dan *UI update*, sehingga setiap komponen dapat dikembangkan dan dikelola secara independen tanpa mengganggu keseluruhan sistem.



Gambar. 2. Arsitektur Sistem

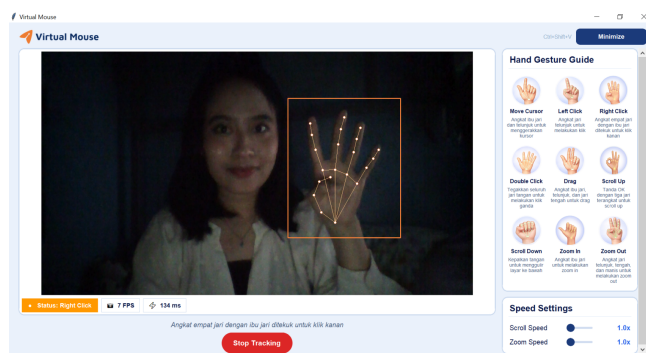
Tahap akhir penelitian ini adalah pengujian sistem melalui dua kategori. *Performance Testing* mencakup *Latency Testing*, *FPS Testing*, dan *Accuracy Testing* yang dijalankan pada dua kondisi perangkat keras (CPU-only dan GPU) untuk mengevaluasi kecepatan dan ketepatan deteksi model. Adapun *Robustness Testing* menjadi fokus utama penelitian ini untuk mengatasi kelemahan penelitian terdahulu terkait ketergantungan pencahayaan, dengan menguji sistem pada kondisi pencahayaan rendah, pencahayaan berlebih, latar belakang kompleks, serta berbagai jarak deteksi guna menentukan rentang optimal penggunaan sistem.

4 Hasil dan Pembahasan

4.1 Analisis Sistem Berjalan dan Rancangan Sistem Usulan

Antarmuka sistem *Virtual Mouse* dibangun menggunakan Tkinter dan CustomTkinter, yang dirancang untuk memberikan umpan balik visual secara *real-time* kepada pengguna selama proses *tracking* berlangsung. Halaman utama terbagi atas beberapa bagian utama. Panel kamera pada sisi kiri menampilkan hasil deteksi tangan berupa *bounding box* dan 21 titik *landmark* yang tervisualisasi langsung di atas gambar kamera, disertai *status pill* berwarna dinamis yang menunjukkan gestur aktif beserta indikator FPS dan latensi secara *real-time*. Panel *Hand*

Gesture Guide pada sisi kanan menyajikan sembilan gestur yang didukung sistem beserta ilustrasi dan deskripsi aksinya, sedangkan panel *Speed Settings* memungkinkan pengguna menyesuaikan kecepatan *scroll* dan *zoom* secara langsung melalui *slider* tanpa perlu menghentikan proses *tracking*. Tampilan antarmuka sistem ditunjukkan pada Gambar 3.



Gambar. 3. Tampilan Antarmuka

Berdasarkan tampilan tersebut, setiap elemen antarmuka dirancang agar pengguna dapat memantau status sistem tanpa perlu mengalihkan perhatian dari aktivitas utama yang sedang dikerjakan. Warna pada *status pill* berubah secara dinamis mengikuti gestur yang terdeteksi, sementara peringatan visual seperti kondisi dua tangan terdeteksi atau tangan kiri yang belum didukung ditampilkan melalui perubahan warna khusus agar pengguna dapat segera menyesuaikan posisinya. Rancangan ini bertujuan menjaga kejelasan umpan balik visual sekaligus mendukung kemudahan penggunaan sistem secara keseluruhan.

4.2 Hasil Pengujian Performa

Pengujian performa dilakukan pada dua kondisi perangkat keras, yaitu CPU-only dan GPU, dengan mengukur tiga parameter utama berupa latensi, FPS, dan akurasi deteksi. Hasil pengujian menunjukkan bahwa GPU memberikan keunggulan signifikan dari sisi kecepatan pemrosesan, dengan latensi rata-rata 54,66 ms dan FPS rata-rata 20,16, yang berhasil memenuhi target *responsiveness* di bawah 100 ms. Sebaliknya, CPU-only menghasilkan latensi rata-rata 239,77 ms dan FPS rata-rata 3,71, yang belum memenuhi target tersebut namun tetap fungsional untuk penggunaan dasar. Pada aspek akurasi deteksi, selisih antara CPU-only sebesar 87,92% dan GPU sebesar 90,65% tergolong tipis, yang menunjukkan bahwa perbedaan perangkat keras tidak memengaruhi kecerdasan model secara signifikan, melainkan hanya memengaruhi kecepatan pemrosesan. Evaluasi lebih lanjut melalui *confusion matrix* menghasilkan nilai *precision* 99,29%, *recall* 99,89%, dan *F1-score* 99,59%, yang mengonfirmasi bahwa model YOLO11 memiliki performa deteksi tangan yang sangat baik dan seimbang tanpa bias signifikan terhadap salah satu kelas.

4.3 Hasil Pengujian Ketahanan

Pengujian ketahanan sistem dilakukan pada tiga skenario utama, yaitu kondisi pencahayaan, latar belakang kompleks, dan jarak deteksi. Pada pengujian kondisi pencahayaan, seluruh sembilan gestur berhasil dikenali dengan status *Pass* tanpa ditemukan kegagalan deteksi pada ketiga rentang intensitas cahaya yang diuji, yaitu pencahayaan rendah, normal, dan berlebih, yang menunjukkan bahwa model memiliki kemampuan adaptasi yang baik terhadap perubahan intensitas cahaya berkat strategi *Random HSV Augmentation* yang diterapkan selama pelatihan. Pengujian pada latar belakang kompleks juga menunjukkan hasil yang konsisten, di mana sistem tetap mampu mendeteksi gestur tangan dengan *bounding box* dan titik *landmark* yang terpetakan secara akurat meskipun terdapat elemen visual yang ramai di sekitar objek tangan, yang membuktikan efektivitas *Copy-Paste Augmentation* dalam meningkatkan ketahanan model terhadap variasi latar belakang. Pada pengujian jarak deteksi, sistem menunjukkan performa terbaik pada jarak 30 cm dengan seluruh gestur berhasil dikenali dan

confidence score tertinggi, performa tetap stabil pada jarak 50 cm dan 70 cm dengan *confidence score* yang masih tinggi, namun mengalami penurunan signifikan pada jarak 100 cm di mana hanya sebagian kecil gestur yang masih dapat dikenali akibat *bounding box* tangan yang mengecil relatif terhadap bingkai kamera. Berdasarkan hasil ini, sistem dapat disimpulkan bekerja optimal pada rentang jarak 30 sampai 70 cm.

4.3 Hasil Pengujian Fungsional dan Penerimaan Pengguna

Pengujian fungsional yang melibatkan 200 *test case* menghasilkan tingkat keberhasilan sebesar 99%, dengan hanya dua kegagalan yang terjadi pada skenario *double click* dan *drag* akibat kesulitan pengguna mempertahankan posisi kursor secara presisi selama gestur berlangsung. Pengujian penerimaan pengguna yang melibatkan 10 penguji menghasilkan penilaian pengalaman pengguna rata-rata sebesar 3,99 dari skala 5,00 dengan kategori Baik, di mana aspek kejelasan umpan balik visual memperoleh penilaian tertinggi, sementara aspek kecepatan respons memperoleh penilaian terendah. Pada evaluasi kebermanfaatan aplikasi, seluruh penguji sepakat bahwa sistem mampu membedakan gestur dengan baik dan layak menjadi alternatif pengganti mouse fisik maupun *touchpad* dalam kegiatan sehari-hari.

4.3 Pembahasan

Secara keseluruhan, hasil pengujian membuktikan bahwa integrasi model YOLO11 dengan logika interpretasi gestur berbasis *landmark* tangan berhasil menghasilkan sistem *Virtual Mouse* yang fungsional dan andal. Perbandingan kinerja pada CPU dan GPU menunjukkan bahwa GPU memberikan keunggulan signifikan dari sisi kecepatan pemrosesan *frame* dan rendahnya latensi, sementara akurasi deteksi model relatif sebanding pada kedua jenis perangkat keras, sehingga perbedaan spesifikasi perangkat keras tidak memengaruhi kecerdasan model secara berarti. Ketahanan sistem terhadap kondisi lingkungan yang menantang, seperti variasi pencahayaan dan latar belakang kompleks, membuktikan bahwa penerapan strategi augmentasi data selama pelatihan efektif mengatasi kelemahan utama yang teridentifikasi pada penelitian-penelitian terdahulu, khususnya ketergantungan terhadap kondisi pencahayaan pada pendekatan berbasis CNN. Meskipun demikian, keterbatasan performa pada perangkat tanpa dukungan GPU serta penurunan tingkat keberhasilan deteksi pada jarak di atas 70 cm menunjukkan bahwa sistem masih memiliki ruang untuk pengembangan lebih lanjut, khususnya pada aspek kecepatan respons yang menjadi nilai terendah dalam penilaian pengalaman pengguna.

5 Kesimpulan dan Saran

5.1 Kesimpulan

Penelitian ini berhasil mengimplementasikan sistem *Virtual Mouse* berbasis *Hand Landmark Detection* menggunakan model YOLO11, yang mendeteksi *bounding box* tangan sekaligus titik landmark dalam satu kali inferensi dan menerjemahkannya menjadi sembilan aksi *mouse* melalui logika geometris. Pengujian fungsional dan *User Acceptance Testing* menunjukkan seluruh aksi berjalan sesuai gestur dan sistem diterima baik oleh pengguna sebagai alternatif *mouse* fisik dan *touchpad*. Dari sisi performa, sistem memenuhi kebutuhan *real-time* pada perangkat GPU dengan kecepatan rata-rata 20,16 FPS dan latensi 54,66 ms, sementara pada CPU-only sistem tetap fungsional meski lebih lambat dengan kecepatan rata-rata 3,71 FPS dan latensi 239,77 ms. Model yang dilatih dengan augmentasi Random HSV dan *Copy-Paste* juga terbukti tahan terhadap pencahayaan rendah, pencahayaan berlebih, dan latar belakang kompleks, serta stabil pada rentang pencahayaan 200 hingga 500 lux. Secara keseluruhan, penelitian ini menjawab seluruh rumusan masalah dan membuktikan bahwa sistem *Virtual Mouse* berbasis YOLO11 merupakan solusi interaksi komputer berbasis gestur tangan yang fungsional, andal, dan dapat diterima pengguna.

5.2 Saran

Sistem *Virtual Mouse* yang dikembangkan masih dapat ditingkatkan melalui beberapa saran berikut. Pertama, penambahan aksi kendali volume suara, seperti memperbesar dan memperkecil volume secara langsung melalui gestur tangan, dapat dipertimbangkan untuk memperluas cakupan aksi yang didukung sistem sehingga pengalaman navigasi komputer tanpa sentuh menjadi lebih lengkap dan fungsional. Kedua, penambahan dukungan untuk tangan kiri perlu dipertimbangkan agar sistem dapat digunakan oleh pengguna yang dominan menggunakan tangan kiri, sehingga aksesibilitas sistem menjadi lebih inklusif. Ketiga, pengembangan dataset yang lebih beragam dengan mencakup variasi kondisi lingkungan, jarak penggunaan, serta karakteristik pengguna yang lebih luas dapat dilakukan untuk meningkatkan kemampuan generalisasi model dalam mengenali gestur tangan pada kondisi penggunaan nyata yang lebih beragam. Dengan mempertimbangkan saran-saran tersebut, diharapkan pengembangan sistem *Virtual Mouse* di masa mendatang dapat menghasilkan solusi yang semakin canggih, inklusif, dan siap diterapkan secara luas sebagai alternatif navigasi komputer yang ergonomis dan efisien.

6 Referensi

- [1] T. A. Berutu, D. L. R. Sigalingging, G. K. V. Simanjuntak, and F. Siburian, "Pengaruh Teknologi Digital terhadap Perkembangan Bisnis Modern," *Neptunus: Jurnal Ilmu Komputer dan Teknologi Informasi*, vol. 2, no. 3, p. 1, 2024, doi: 10.61132/neptunus.v2i3.258.
- [2] D. Waru and R. W. Astuti, "Implementasi Metode Naive Bayes Untuk Prediksi Daerah Terjangkit Demam Berdarah Dengue Di Provinsi Jambi," *JTIK (Jurnal Teknik Informatika Kaputama)*, vol. 5, no. 2, pp. 240–245, 2021, doi: 10.59697/jtik.v5i2.550.
- [3] J. Ken, T. Darmanto, and H. Hendro, "Rancang Bangun Sistem Virtual Mouse Dengan Hand Gesture Recognition Menggunakan Convolutional Neural Network," vol. 12, no. 1, pp. 1–9, 2025, doi: 10.5281/ZENODO.15621182.
- [4] N. Dhengre, N. S. Rajput, V. Patel, and R. Katarne, "A Perspective of the Computer Mouse's Impact on the Musculoskeletal System," *Evergreen*, vol. 11, no. 3, p. 1, 2024, doi: 10.5109/7236843.
- [5] K. Hernando, "Implementasi Deep Learning Untuk Visi Komputer Sebagai Mouse Virtual Menggunakan Mediapipe Dan Faster RCNN," 2023, pp. 30–37.
- [6] S. Karthick, D. M. J. D. Raj C, and N. Jayapandian, "Artificial Intelligence Based Enhanced Virtual Mouse Hand Gesture Tracking Using Yolo Algorithm," in *2023 IEEE 2nd International Conference on Data, Decision and Systems (ICDDS)*, 2023, pp. 1–6, doi: 10.1109/ICDDS59137.2023.10434330.
- [7] M. Krishnamoorthi, S. Gowtham, K. Sanjeevi, and R. Revanth Vishnu, "Virtual Mouse Using YOLO," in *2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 2022, pp. 1–7, doi: 10.1109/ICSES55317.2022.9914153.
- [8] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, "YOLO Evolution: A Comprehensive Benchmark and Architectural Review of YOLOv12, YOLO11, and Their Previous Versions," version 4, arXiv, 2024, doi: 10.48550/ARXIV.2411.00201.
- [9] R. Riswan and N. Nilawati, *Interaksi Manusia dan Komputer: Teori, Metode dan Aplikasi*. PT. Star Digital Publishing, 2025.
- [10] M. Z. Iqbal and A. G. Campbell, "From Luxury to Necessity: Progress of Touchless Interaction Technology," *Technology in Society*, vol. 67, pp. 1–2, 2021, doi: 10.1016/j.techsoc.2021.101796.

- [11] R. Annachhatre, M. Tamakuwala, P. Shinde, A. Jain, and P. V. Kulkarni, "Virtual Mouse Using Hand Gesture Recognition—A Systematic Literature Review," in *2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*, vol. 11, 2022, pp. 1–6, doi: 10.1109/ICBDS53701.2022.9935944.
- [12] F. H. Laia, R. Rosnelly, A. Naswar, K. Buulolo, and M. C. M. Lase, "Deteksi Pengenalan Wajah Orang Berbasis AI Computer Vision," *Jurnal Teknologi Informasi Mura*, vol. 15, no. 1, pp. 1–9, 2023, doi: 10.32767/jti.v15i1.2024.
- [13] A. R. Baehaqi, "Implementasi Object Detection Pada Kasus Pelanggaran Rambu Dilarang Belok Dengan Menggunakan Algoritma Yolo," Universitas Komputer Indonesia, 2021.
- [14] C. Zhao, T. Wu, J. Zhang, Z. Xu, M. Li, and D. Liu, "Siamese Comparative Transformer-Based Network for Unsupervised Landmark Detection," *PLOS ONE*, vol. 19, no. 12, pp. 1–6, 2024, doi: 10.1371/journal.pone.0313518.
- [15] J. P. Sahoo, A. J. Prakash, P. Pławiak, and S. Samantray, "Real-Time Hand Gesture Recognition Using Fine-Tuned Convolutional Neural Network," *Sensors*, vol. 22, no. 3, pp. 1–4, 2022, doi: 10.3390/s22030706.
- [16] Y. Bengio, Y. Lecun, and G. Hinton, "Deep Learning for AI," *Communications of the ACM*, vol. 64, no. 7, pp. 58–65, 2021, doi: 10.1145/3448250.
- [17] M. Tariq and K. Choi, "YOLO11-Driven Deep Learning Approach for Enhanced Detection and Visualization of Wrist Fractures in X-Ray Images," *Mathematics*, vol. 13, no. 9, pp. 2–9, 2025, doi: 10.3390/math13091419.
- [18] Y. Pratama, F. T. Wulandari, and W. Sanjaya, "Implementasi Metode Rapid Application Development pada Sistem Informasi Harga Barang Berbasis Web di Perusahaan Cipta Warna Digital Printing Boyolali," *JITU: Journal Informatic Technology and Communication*, vol. 7, no. 2, p. 2, 2023, doi: 10.36596/jitu.v7i2.794.