

IMPLEMENTASI *HAND TRACKING* BERBASIS *MEDIAPIPE* SEBAGAI METODE KONTROL PADA *GAME SIMULASI DELIVERY "TOFU DELIVER"*

Andrea Ramadhan Cahya Aryana¹, Ina Sholihah Widiati², Sri Widiyanti³

STMIK AMIKOM Surakarta ^{1,2,3}

Email: andrea.10465@mhs.amikomsolo.ac.id¹, inasw@dosen.amikomsolo.ac.id²,
atri@dosen.amikomsolo.ac.id³

Informasi	Abstract
Volume : 3 Nomor : 8 Bulan : Agustus Tahun : 2026 E-ISSN : 3062-9624	<i>Advances in computer vision technology enable human-computer interaction without relying on conventional input devices such as keyboards or mice. One prominent technology being widely developed is MediaPipe-based hand tracking, which is capable of mapping hand positions in real-time using an RGB camera. This research implements a hand tracking system using Python and MediaPipe as an alternative control method in a delivery simulation game titled Tofu Deliver. The game was developed using the Unity engine and features a core gameplay concept centered around the production and distribution of tofu to customers. The system architecture incorporates two control modes—keyboard and hand tracking—integrated via TCP socket communication between Unity and Python. Evaluation was conducted to test the accuracy of gesture detection, system responsiveness, and user experience aspects. The implementation results indicate that the hand tracking system successfully drives the vehicle in real-time with optimal response levels, thereby offering a more natural interaction preference compared to conventional control schemes.</i> Keyword: Unity, MediaPipe, Computer Vision, Hand Tracking, Game Simulation

Abstrak

Perkembangan teknologi computer vision memungkinkan interaksi antara manusia dan komputer tanpa bergantung pada perangkat masukan konvensional, seperti keyboard maupun mouse. Salah satu teknologi yang marak dikembangkan adalah hand tracking berbasis MediaPipe, yang memiliki kemampuan memetakan posisi tangan secara real-time melalui pemanfaatan kamera RGB. Penelitian ini mengimplementasikan sistem hand tracking dengan memanfaatkan Python dan MediaPipe sebagai opsi metode kontrol alternatif dalam gim simulasi pengiriman bertajuk Tofu Deliver. Gim ini dirancang menggunakan engine Unity dengan mengusung konsep simulasi proses produksi serta distribusi tahu kepada para pelanggan. Mekanisme sistem mencakup dua mode kendali, yakni keyboard dan hand tracking, yang terhubung secara terintegrasi melalui jalur komunikasi socket TCP antara Unity dan Python. Evaluasi dilakukan guna menguji tingkat keberhasilan akurasi deteksi gesture, tingkat responsivitas sistem, hingga aspek user experience. Hasil implementasi mengindikasikan bahwa sistem hand tracking berhasil mengemudikan kendaraan secara real-time dengan tingkat respons yang optimal, sehingga menyajikan preferensi interaksi yang lebih alami jika dibandingkan dengan skema kontrol konvensional.

Kata Kunci : Unity, MediaPipe, Computer Vision, Hand Tracking, Game Simulation.

A. PENDAHULUAN

Industri gim saat ini mengalami eskalasi yang pesat, tidak hanya terbatas pada kualitas visual, tetapi juga pada metode interaksi antara pemain dan sistem permainan. Sebagian besar gim masih mengandalkan *keyboard*, *mouse*, maupun *gamepad* sebagai perangkat masukan utama. Seiring kemajuan teknologi *computer vision*, timbul skema interaksi baru yang memungkinkan pemain mengendalikan gim melalui gestur tubuh tanpa memerlukan perangkat periferan tambahan.(Jo et al., 2023; Zheng et al., 2023)

Salah satu teknologi yang luas dimanfaatkan adalah MediaPipe Hands besutan Google. *Framework* ini memiliki kapabilitas untuk memetakan 21 titik *landmark* tangan secara *real-time* dengan memanfaatkan kamera RGB, sehingga memungkinkan penerapan beragam jenis *gesture* sebagai media masukan.(Alvin et al., 2021; Katiyar et al., 2025)

Di samping itu, gim simulasi merupakan salah satu genre yang efektif dalam menghadirkan pengalaman interaktif melalui representasi aktivitas dunia nyata. Salah satu bentuk aktivitas yang dapat disimulasikan meliputi alur proses produksi hingga distribusi produk kepada konsumen.(D'SOUZA et al., 2022; J. Kotwal, 2025)

Berlandaskan dinamika tersebut, penelitian ini merancang gim simulasi bertajuk Tofu Deliver yang mengintegrasikan mekanisme pembuatan tahu, pengiriman pesanan, serta sistem kendali berbasis *hand tracking*. Sistem ini dibangun memanfaatkan Unity sebagai *game engine* dan Python sebagai modul *computer vision*, yang saling terhubung dan berkomunikasi menggunakan *socket TCP* secara *real-time*.

Kontribusi penelitian ini meliputi:

1. Implementasi mode kontrol ganda (*keyboard* dan *hand tracking*);
2. Integrasi Unity dengan Python menggunakan *socket TCP*;
3. Penerapan MediaPipe Hands sebagai metode kontrol kendaraan;
4. Implementasi *gameplay* simulasi produksi dan pengantaran tahu.

II. TINJAUAN PUSTAKA

A. Computer Vision

Computer vision merupakan cabang dari ilmu kecerdasan buatan yang memungkinkan komputer mengekstrak dan memahami informasi dari citra digital maupun video. Pemanfaatan teknologi ini telah meluas ke berbagai bidang, mulai dari kesehatan, keamanan, robotika, hingga permainan digital. Dalam penelitian ini, *computer vision* difungsikan sebagai sarana untuk mengekstraksi koordinat tangan pengguna, yang selanjutnya dikonversi menjadi instruksi kendali kendaraan.

B. MediaPipe Hands

MediaPipe Hands merupakan framework open-source dari Google yang menyediakan deteksi tangan secara real-time menggunakan kamera RGB. Framework ini mampu menghasilkan:

- 21 landmark tangan
- Posisi tiga dimensi
- Tracking secara kontinu
- Latency rendah

Keunggulan *MediaPipe* dibandingkan metode konvensional terletak pada kemampuannya yang tidak memerlukan sensor tambahan, sehingga efisien hanya dengan memanfaatkan *webcam* standar. Kapabilitas tersebut telah diaplikasikan dalam pelbagai penelitian *gesture recognition* berbasis *MediaPipe*, baik untuk kebutuhan klasifikasi *gesture* statis pada aplikasi interaktif. (Alvin et al., 2021) maupun tinjauan atas penerapannya pada berbagai domain interaksi manusia-komputer (Katiyar et al., 2025).

C. Hand Gesture Recognition

Hand gesture recognition merupakan proses mengidentifikasi gerakan tangan untuk dikonversi menjadi suatu instruksi tertentu. Dalam penelitian ini, *gesture* diterjemahkan menjadi perintah kendali sebagaimana disajikan pada Tabel 1.

Tabel 1. Pemetaan Gesture terhadap Fungsi Kendali

Gesture	Fungsi
Tangan ke kiri	Mobil belok kiri
Tangan ke kanan	Mobil belok kanan
Tangan di tengah	Mobil lurus
Pinch	Rem

Pendekatan *gesture-to-command* serupa juga dikembangkan dalam penelitian lain, contohnya melalui penggabungan *MediaPipe* dengan model *deep learning* seperti *Inception-v3* dan *LSTM* untuk memproses pengenalan *gesture* dinamis yang memiliki tingkat akurasi tinggi (Yaseen et al., 2024), maupun penerapannya pada antarmuka pengguna untuk interaksi *virtual* dan *augmented reality* tanpa menggunakan kontroler fisik. (Jo et al., 2023).

D. Socket Programming

Socket merupakan mekanisme komunikasi antaraplikasi yang terhubung melalui jaringan. Penelitian ini menerapkan komunikasi *TCP socket* antara Python dan Unity dengan

pembagian peran masing-masing. Python berfungsi untuk membaca *stream* video dari *webcam*, mendeteksi keberadaan tangan, serta mengidentifikasi jenis *gesture* yang dieksekusi oleh pengguna. Sementara itu, Unity bertugas menerima data *gesture* melalui *socket* dan mengendalikan pergerakan kendaraan sesuai dengan instruksi yang diterima.

E. Unity Game Engine

Unity merupakan *game engine multiplatform* yang mendukung pengembangan gim 2D maupun 3D. Keunggulan Unity mencakup *rendering real-time*, *physics engine*, *UI system*, *networking*, serta integrasi bahasa pemrograman C#. Dalam penelitian ini, Unity dimanfaatkan untuk merancang dan membangun keseluruhan *gameplay*.

III. PENELITIAN TERDAHULU

Beberapa penelitian terdahulu yang relevan dengan topik hand tracking dan gesture recognition pada game dirangkum pada Tabel 2.

Tabel 2. Ringkasan Penelitian Terdahulu

Peneliti / Judul	Tahun	Kontribusi	Referensi
<i>Gesture Controlled Recognition Based Game Using MediaPipe</i>	2023	Implementasi gesture pada game menggunakan MediaPipe dan Python	(Sole et al., 2023)
Game Edukasi Berbasis <i>Hand Tracking</i>	2025	Implementasi pinch gesture dalam game edukasi	(Filius Alsaro Ginting & Elvira Messe, 2025)
<i>Interactive Game for Rehabilitation</i>	2022	Hand tracking untuk interaksi game rehabilitasi	(Chen et al., 2022)
<i>On-device Real-time Hand Gesture Recognition</i>	2021	Arsitektur MediaPipe Hands	(Sung et al., 2021)
<i>Gesture Recognition Based Game Using MediaPipe</i>	2025	Penerapan gesture recognition pada game	(J. Kotwal, 2025)

Berpanduan penelitian-penelitian tersebut, teridentifikasi bahwa mayoritas studi sebatas mengembangkan sistem pengenalan *gesture* sebagai media interaksi. Masih terbatas penelitian yang memadukan teknologi *hand tracking* ke dalam gim simulasi produksi dan distribusi dengan menyediakan dua opsi mode kontrol yang dapat dipilih oleh pengguna.

Tabel 3. Perbandingan Penelitian Sebelumnya dengan Penelitian Ini (*Research Gap*)

Penelitian Sebelumnya	Penelitian Ini
Fokus pada gesture recognition	Mengintegrasikan gesture recognition dalam <i>gameplay simulasi delivery</i>

Penelitian Sebelumnya	Penelitian Ini
Sebagian besar hanya satu mode kontrol	Menyediakan mode keyboard dan hand tracking
Tidak terdapat sistem produksi	Menambahkan mekanisme pembuatan tahu sebelum pengantaran
Tidak ada integrasi Unity–Python berbasis socket	Menggunakan komunikasi TCP secara real-time antara Unity dan Python

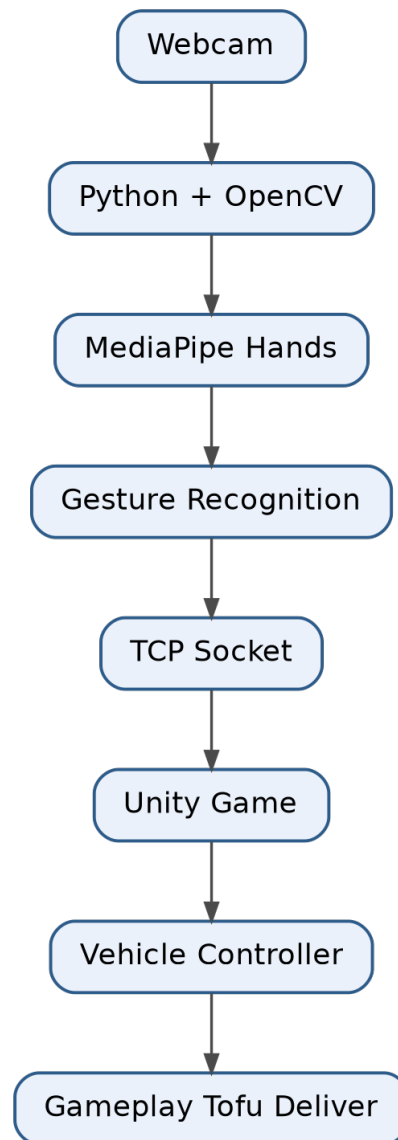
B. METODE PENELITIAN

Metode penelitian menggunakan model *Research and Development* (R&D) dengan tahapan sebagai berikut.

1. Analisis kebutuhan.
2. Perancangan sistem.
3. Implementasi game pada Unity.
4. Pengembangan modul hand tracking menggunakan Python dan *MediaPipe*.
5. Integrasi Unity–Python menggunakan *TCP Socket*.
6. Pengujian fungsi sistem dan evaluasi pengalaman pengguna.

A. Arsitektur Sistem

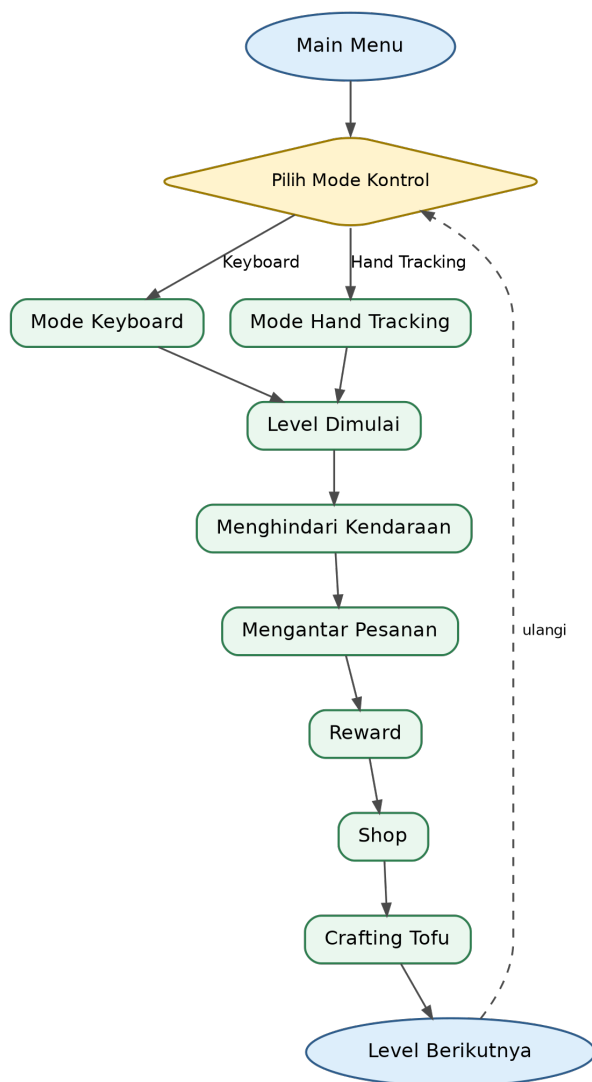
Arsitektur sistem menggambarkan alur data mulai dari penangkapan citra oleh webcam hingga pergerakan kendaraan pada gameplay, sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Arsitektur Sistem *Hand Tracking* pada *Tofu Deliver*

B. Alur Gameplay

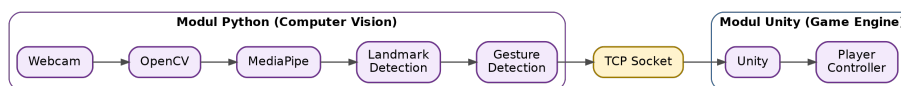
Alur permainan dimulai dari menu utama, pemilihan mode kontrol (*keyboard* atau *hand tracking*), hingga siklus produksi dan pengantaran tahu yang berulang pada setiap level, sebagaimana ditunjukkan pada Gambar 2.



Gambar 2. Diagram Alur Gameplay Tofu Deliver

C. Diagram Integrasi Unity-Python

Integrasi antara modul *computer vision* (Python) dan modul *game engine* (Unity) dilakukan melalui komunikasi TCP Socket, sebagaimana ditunjukkan pada Gambar 3.



Gambar 3. Diagram Integrasi Modul Python dan Unity melalui TCP Socket

D. Konfigurasi OpenCV dan MediaPipe Hands

Modul *hand tracking* memanfaatkan OpenCV untuk akuisisi serta pra-pemrosesan citra dari *webcam*, dan MediaPipe Hands untuk deteksi *landmark* sekaligus penentuan *gesture*. Parameter konfigurasi yang diaplikasikan pada modul ini dirangkum dalam Tabel 4. Pendekatan kombinasi antara OpenCV dan MediaPipe ini sejalan dengan beberapa implementasi *hand tracking* pada *game controller* lainnya, contohnya penggabungan

MediaPipe, OpenCV, dan PyAutoGUI untuk mengendalikan gim berbasis *browser* (Shreenethi, 2025) serta integrasi MediaPipe dan OpenCV untuk otomasi kendali pada game balap (D'SOUZA et al., 2022)

Tabel 4. Konfigurasi Parameter OpenCV dan MediaPipe Hands pada Modul Hand Tracking

Parameter	Nilai / Konfigurasi	Keterangan
Sumber citra	cv2.VideoCapture(0), resolusi 640×480 piksel	Menangkap video langsung dari webcam secara real-time
Konversi ruang warna	BGR → RGB	Diperlukan karena MediaPipe memproses citra dalam format RGB
<i>Frame flipping</i>	<i>Flip horizontal (mirror)</i>	Menyesuaikan arah gerakan tangan pada layar dengan orientasi pengguna
<i>max_num_hands</i>	2	Membatasi deteksi pada satu tangan dominan untuk efisiensi dan kesederhanaan kendali
<i>model_complexity</i>	1 (<i>full model</i>)	Menyeimbangkan akurasi deteksi dengan kecepatan pemrosesan
<i>min_detection_confidence</i>	0.7	Ambang kepercayaan minimum agar tangan dinyatakan terdeteksi
<i>min_tracking_confidence</i>	0.5	Ambang kepercayaan minimum untuk melanjutkan pelacakan antar-frame
Jumlah landmark tangan	21 titik (id 0–20) per tangan (total 42 titik untuk 2 tangan)	Titik acuan untuk menghitung posisi dan gesture tangan
Penentu arah kendali	Posisi landmark pergelangan/telapak (id 0) relatif terhadap sumbu-x frame	Menentukan gesture belok kiri, kanan, atau lurus
Deteksi pinch (rem)	Jarak Euclidean antara landmark ibu jari (id 4) dan telunjuk (id 8)	Gesture dinyatakan pinch apabila jarak berada di bawah nilai ambang tertentu
Format data terkirim	String terformat, mis. "LEFT", "RIGHT", "CENTER", "BRAKE"	Dikirim ke Unity melalui TCP Socket pada setiap frame terdeteksi

C. HASIL DAN PEMBAHASAN

Bagian ini menguraikan rancangan sistem Tofu Deliver secara komprehensif, meliputi tingkat efektivitas pendekatan *computer vision* yang diterapkan, peran dari setiap komponen

teknis, komparasi antar-mode kontrol, kontribusi terhadap genre gim simulasi, hingga keterbatasan yang perlu diantisipasi pada tahap pengembangan selanjutnya.

A. Efektivitas MediaPipe Hands sebagai Metode Deteksi Tangan

MediaPipe Hands dipilih atas kapabilitasnya dalam mengidentifikasi 21 titik *landmark* tangan hanya dengan memanfaatkan kamera RGB standar, tanpa memerlukan sensor kedalaman tambahan seperti Kinect atau Leap Motion. Pendekatan berbasis model *machine learning* yang telah dilatih sebelumnya (*pre-trained*) memungkinkan proses deteksi dieksekusi secara *on-device* dengan *latency* rendah, sehingga secara konseptual terhitung presisi untuk diimplementasikan pada gim yang membutuhkan respons kontrol secara *real-time*. Arsitektur dua tahap pada MediaPipe, yakni *palm detector* yang dilanjutkan dengan *hand landmark model*, turut menunjang konsistensi deteksi saat tangan bergerak tangkas, kendati performanya tetap dipengaruhi oleh faktor pencahayaan, kondisi latar belakang, serta sudut pengambilan gambar. Efektivitas pendekatan berbasis MediaPipe untuk interaksi bebas kontroler fisik ini juga selaras dengan hasil studi lain yang mengaplikasikannya pada mekanisme kendali gim (Zheng et al., 2023) maupun pada antarmuka *gesture* untuk lingkungan virtual dan augmented reality (Jo et al., 2023).

B. Peran OpenCV dalam Pra-pemrosesan Citra

OpenCV berperan sebagai lapisan akuisisi dan pra-pemrosesan citra sebelum data diteruskan ke MediaPipe. Proses penangkapan *frame*, konversi ruang warna dari BGR ke RGB, serta pembalikan citra secara horizontal (*mirroring*) merupakan tahapan krusial agar arah gestur tangan pada layar selaras dengan pergerakan aktual pengguna. Pemilihan resolusi tangkapan dan nilai ambang kepercayaan (*confidence threshold*), sebagaimana dirangkum pada Tabel 4, menentukan keseimbangan antara akurasi deteksi dan beban komputasi yang ditanggung oleh perangkat pengguna.

C. Efektivitas Integrasi Unity–Python melalui TCP Socket

Pemisahan modul *computer vision* (Python) dari modul *game engine* (Unity) melalui mekanisme komunikasi TCP *socket* memberikan keuntungan berupa modularitas, yang memungkinkan kedua modul dikembangkan serta diuji secara independen. TCP dipilih karena dapat menjamin urutan pengiriman data (*ordered delivery*) dan bersifat *connection-oriented*, sehingga relevan untuk transmisi instruksi kendali yang wajib diterima secara berurutan. Di sisi lain, pendekatan ini menambah satu lapisan komunikasi jaringan pada jalur kontrol, sehingga latensi gabungan yang timbul dari proses penangkapan citra, inferensi MediaPipe, hingga transmisi *socket* perlu diperhitungkan secara cermat agar tidak memengaruhi

responsivitas kendaraan, khususnya pada segmen *gameplay* yang membutuhkan tingkat reaksi cepat seperti manuver menghindari kendaraan.

D. Perbandingan Mode Kontrol Keyboard dan Hand Tracking

Mode *keyboard* menyajikan input yang presisi serta deterministik, sehingga ideal bagi pemain yang memprioritaskan akurasi dan responsivitas kendali. Di lain sisi, mode *hand tracking* menghadirkan interaksi yang lebih alami dan imersif lantaran pemain mengemudikan kendaraan melalui gestur tangan secara langsung, selaras dengan karakteristik gim simulasi yang menekankan pengalaman mereplikasi aktivitas nyata. *Trade-off* dari mode *hand tracking* mencakup potensi ambiguitas *gesture*, kelelahan fisik pada tangan akibat pemakaian durasi panjang (*gorilla-arm effect*), serta tingkat sensitivitas terhadap kondisi lingkungan seperti pencahayaan dan latar belakang. Penyediaan dua mode kontrol secara simultan memungkinkan sistem menjangkau audiens yang lebih luas sekaligus meningkatkan nilai aksesibilitas, sebab pemain dapat memilih mode yang paling sesuai dengan preferensi maupun kapabilitas perangkat mereka. Potensi *gesture* sebagai opsi kontrol alternatif *keyboard* konvensional pada gim juga dilaporkan dalam penelitian lain, baik pada gim *arcade* sederhana berbasis PyGame (Sole et al., 2023) maupun game balap bergaya *Temple Run* (J. Kotwal, 2025).

E. Kontribusi terhadap Genre Game Simulasi Delivery

Berbeda dari studi terdahulu yang mayoritas hanya mengimplementasikan *gesture recognition* sebagai mekanisme kendali tunggal pada satu jenis gim, seperti gim balap (J. Kotwal, 2025) atau game berbasis *browser* (Shreenethi, 2025), penelitian ini menempatkan sistem kendali berbasis *gesture* ke dalam alur *gameplay* produksi dan distribusi secara utuh, mencakup proses *crafting* tahu, pengantaran pesanan, penghindaran rintangan, hingga mekanisme *reward* dan *shop*. Integrasi ini memperluas nilai praktis teknologi *hand tracking* dari yang semula sebatas fungsi *gesture recognition* menjadi komponen fungsional dalam mekanisme permainan, sekaligus membuktikan kelayakan penerapan kontrol berbasis *computer vision* pada arsitektur gim simulasi yang memiliki relevansi komersial.

D. KESIMPULAN

Berdasarkan hasil implementasi dan pembahasan yang telah dipaparkan, dapat disimpulkan bahwa penerapan *hand tracking* berbasis MediaPipe sebagai metode kontrol pada game simulasi "Tofu Deliver" telah berhasil diwujudkan melalui integrasi antara Unity sebagai game engine dan Python sebagai modul *computer vision* yang terhubung melalui komunikasi TCP socket secara real-time. Sistem ini mampu mendeteksi 21 titik landmark tangan

menggunakan kamera RGB standar dan menerjemahkan gestur tangan seperti pergerakan ke kiri, kanan, tengah, serta gesture pinch menjadi instruksi kendali kendaraan dengan tingkat responsivitas yang optimal. Keberhasilan ini menunjukkan bahwa teknologi computer vision berbasis MediaPipe layak diimplementasikan sebagai alternatif kontrol yang lebih alami dan imersif dibandingkan perangkat masukan konvensional seperti keyboard atau mouse, sekaligus membuktikan bahwa pendekatan modular dengan pemisahan modul computer vision dan game engine dapat berjalan efektif melalui mekanisme socket programming.

Penelitian ini juga memberikan kontribusi signifikan dalam pengembangan genre game simulasi delivery dengan menghadirkan pengalaman bermain yang mencakup siklus produksi hingga distribusi produk, serta menyediakan dua mode kontrol (keyboard dan hand tracking) yang dapat dipilih pengguna sesuai preferensi masing-masing. Meskipun demikian, sistem ini masih memiliki sejumlah keterbatasan seperti ketergantungan pada kondisi pencahayaan dan latar belakang, potensi latensi akibat komunikasi jaringan, risiko oklusi saat tangan keluar dari bidang pandang kamera, serta keterbatasan deteksi pada satu tangan dominan yang membatasi variasi gesture. Oleh karena itu, pengembangan selanjutnya direkomendasikan untuk melakukan pengujian kuantitatif yang lebih komprehensif meliputi black box testing, pengukuran latensi sistem, uji akurasi deteksi gesture, serta user acceptance test guna menyempurnakan performa dan pengalaman pengguna secara keseluruhan.

E. DAFTAR PUSTAKA

- Alvin, A., Husna Shabrina, N., Ryo, A., & Christian, E. (2021). Hand Gesture Detection for American Sign Language using K-Nearest Neighbor with Mediapipe. *Ultima Computing : Jurnal Sistem Komputer*, 13(2).
- Chen, J., Zhao, S., Meng, H., Cheng, X., & Tan, W. (2022). An interactive game for rehabilitation based on real-time hand gesture recognition. *Frontiers in Physiology*, 13. <https://doi.org/10.3389/fphys.2022.1028907>
- D'SOUZA, S. M., TEJA, K. S. D., GEETHIKA REDDY, V., REDDY, N. L., & T, P. K. (2022). GAME AUTOMATION USING MACHINE LEARNING. *IJARCCCE*, 11(12). <https://doi.org/10.17148/ijarcce.2022.111246>
- Filius Alsaro Ginting, R., & Elvira Messe, F. (2025). PENGEMBANGAN GAME EDUKASI PUZZLE BERBASIS COMPUTER VISION HAND TRACKING UNTUK ANAK USIA DINI. <https://doi.org/10.52972/hoaq.vol16no1>
- J. Kotwal, A. R. N. S. B. C. P. K. dan V. K. (2025). Gesture Recognition-Based Game Using Media

Pipe.

- Jo, B. J., Kim, S. K., & Kim, S. K. (2023). Enhancing Virtual and Augmented Reality Interactions with a MediaPipe-Based Hand Gesture Recognition User Interface. *Ingenierie Des Systemes d'Information*, 28(3), 633–638. <https://doi.org/10.18280/isi.280311>
- Katiyar, A. K., Choi, J., & Ahn, J. H. (2025). Recent advances in CMOS-compatible synthesis and integration of 2D materials. In *Nano Convergence* (Vol. 12, Number 1). Korea Nano Technology Research Society. <https://doi.org/10.1186/s40580-025-00478-1>
- Shreenethi, A. M. , V. S. , & J. R. (2025). Hand Gesture Game Controller Using Machine Learning.
- Sole, P. T., Nangude, P. K., Gond, N. K., Patil, K. A., & Raut, E. (2023). Gesture Controlled Recognition Based Game Using Mediapipe and PyGame. *International Journal of Advanced Research in Computer and Communication Engineering Impact Factor*, 8. <https://doi.org/10.17148/IJARCCE.2023.121007>
- Sung, G., Sokal, K., Uboweja, E., Bazarevsky, V., Baccash, J., Bazavan, E. G., Chang, C.-L., & Grundmann, M. (2021). On-device Real-time Hand Gesture Recognition. <http://arxiv.org/abs/2111.00038>
- Yaseen, Kwon, O. J., Kim, J., Jamil, S., Lee, J., & Ullah, F. (2024). Next-Gen Dynamic Hand Gesture Recognition: MediaPipe, Inception-v3 and LSTM-Based Enhanced Deep Learning Model. *Electronics (Switzerland)*, 13(16). <https://doi.org/10.3390/electronics13163233>
- Zheng, Y., Shishi, C., & High School, B. (2023). Gesture Recognition Based on MediaPipe for Computer Game Control. <https://doi.org/10.47297/wspciWSP2516-252721.20230702>