

Pemanfaatan Kamera untuk Kontrol Gerakan Real-Time dalam Sistem Interaktif Berbasis Unreal Engine

Baiq Fadilatul Islamiah, Muh.Farid Hakim, M. Rizqullah, Kurniadin Abdul latif

Universitas Bumigora, Mataram, Indonesia

Correspondence : e-mail: kurniadin@univeritasbumigora.ac.id

Abstrak

Kemajuan teknologi pose estimation telah mendorong pengembangan sistem interaktif yang memanfaatkan gerakan tubuh pengguna sebagai input. Penelitian ini bertujuan untuk mengembangkan sistem kontrol karakter di Unreal Engine yang dikendalikan secara real-time melalui empat jenis gerakan tubuh, yaitu mengangkat tangan kanan, mengangkat tangan kiri, menekuk tangan kiri, dan menarik tangan kiri ke belakang. Sistem menggunakan MediaPipe BlazePose untuk mendeteksi 33 titik tubuh dari citra kamera secara langsung, kemudian mengubahnya menjadi data sudut gerakan (Euler angle) yang dikirim ke Unreal Engine melalui protokol UDP. Hasil implementasi menunjukkan bahwa sistem mampu merespons gerakan dengan cepat dan akurat, serta berjalan stabil pada perangkat tanpa GPU tambahan. Evaluasi kuisioner terhadap 32 responden menunjukkan bahwa sistem ini dinilai menarik, unik, dan berpotensi untuk digunakan dalam konteks game dan simulasi. Meskipun demikian, akurasi sistem masih dipengaruhi oleh pencahayaan dan kecepatan gerakan. Penelitian ini menunjukkan bahwa kontrol gerakan tubuh menggunakan kamera biasa dan algoritma visi komputer dapat menjadi alternatif interaksi manusia-komputer yang efisien dan ekonomis.

Kata kunci: Pose Estimation, Kamera, Real-Time, Unreal Engine, Sistem Interaktif.

Abstract

The advancement of pose estimation technology has driven the development of interactive systems that utilize users' body movements as input. This study aims to develop a character control system in Unreal Engine that operates in real-time through four types of body gestures: raising the right arm, raising the left arm, bending the left elbow, and retracting the left arm backward. The system uses MediaPipe BlazePose to detect 33 body landmarks directly from the camera feed, then converts the data into movement angles (Euler angles), which are transmitted to Unreal Engine via the UDP protocol. The implementation results show that the system can respond to movements quickly and accurately, while also running stably on devices without a dedicated GPU. A questionnaire evaluation involving 32 respondents indicated that the system is perceived as interesting, unique, and has potential for use in gaming and simulation contexts. However, the system's accuracy is still influenced by lighting conditions and the speed of the user's movements. This research suggests that body movement control using a standard camera and computer vision algorithms can serve as an efficient and cost-effective alternative for human-computer interaction.

Keywords: Pose Estimation, Camera, Real-Time, Unreal Engine, Interactive System.

1. Pendahuluan

Perkembangan teknologi digital dan kecerdasan buatan telah membawa dampak besar dalam dunia komputasi interaktif. Salah satu kemajuan yang paling signifikan adalah munculnya teknologi yang memungkinkan interaksi manusia-komputer melalui gerakan tubuh. Interaksi semacam ini memberikan pengalaman yang lebih alami dan intuitif karena menghilangkan kebutuhan akan perangkat input konvensional seperti joystick, mouse, atau keyboard. Salah satu pendekatan paling menonjol dalam bidang ini adalah penggunaan teknik pose estimation, yaitu metode yang digunakan untuk mendeteksi posisi tubuh manusia dari citra visual secara real-time.[1] Teknik pose estimation mampu mendeteksi titik-titik kunci tubuh manusia, seperti pergelangan tangan, siku, bahu, lutut, dan lainnya, yang kemudian dapat digunakan

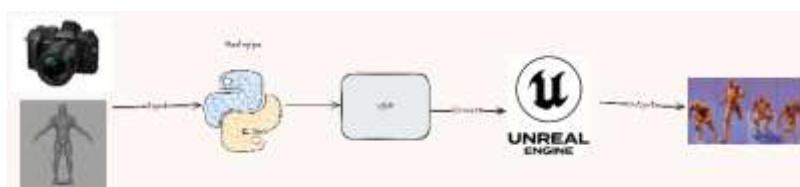
untuk menganalisis atau mengontrol suatu sistem digital. Dengan bantuan kamera biasa dan algoritma visi komputer, sistem ini dapat mengidentifikasi postur dan gerakan pengguna tanpa memerlukan sensor atau penanda tambahan. Pendekatan ini tidak hanya lebih hemat biaya, tetapi juga lebih praktis untuk diimplementasikan pada berbagai platform, termasuk perangkat komputer pribadi dan aplikasi berbasis web.[2]

Dalam dunia pengembangan aplikasi visual dan simulasi, Unreal Engine telah menjadi salah satu platform yang banyak digunakan karena kemampuannya dalam menghasilkan grafik berkualitas tinggi dan dukungan terhadap integrasi teknologi terbaru. Penggabungan antara pose estimation dan Unreal Engine membuka peluang besar untuk membangun sistem interaktif yang mampu merespons gerakan tubuh secara langsung dan akurat. Sistem semacam ini sangat potensial untuk diterapkan dalam bidang seperti pelatihan virtual, pendidikan berbasis simulasi, terapi rehabilitasi, dan permainan digital berbasis gesture.[3] Studi terdahulu telah menunjukkan bahwa integrasi teknologi penangkapan gerakan ke dalam platform seperti Unreal Engine dapat menghasilkan simulasi yang imersif dan responsif. Sebagai contoh, penggunaan Rokoko Motion Capture telah terbukti mampu menangkap gerakan manusia dengan detail tinggi dan menerapkannya secara langsung ke dalam karakter virtual.[4] Namun demikian, pendekatan ini masih menghadapi kendala seperti harga perangkat yang mahal dan kebutuhan ruang khusus untuk pemasangan sensor.

Sebagai solusi dari keterbatasan tersebut, banyak peneliti mulai memanfaatkan metode berbasis visi komputer seperti MediaPipe, yang dikembangkan oleh Google. MediaPipe menyediakan pipeline deteksi gerakan tubuh berbasis machine learning yang dapat bekerja dengan cepat dan ringan, bahkan pada perangkat dengan spesifikasi menengah.[5] Penggunaan MediaPipe untuk mengenali gerakan tangan atau tubuh telah diterapkan dalam berbagai aplikasi interaktif, termasuk game edukasi, terapi motorik anak, dan aplikasi virtual reality. Sebagai ilustrasi, penelitian menunjukkan bagaimana motion capture berbasis kamera dapat digunakan untuk membentuk avatar virtual yang ekspresif dalam dunia metaverse.[5] Studi lain juga menunjukkan bahwa sistem interaksi non-kontak berbasis gesture dapat dirancang dan dievaluasi secara efisien menggunakan kamera biasa dan algoritma computer vision, serta memperoleh tingkat kenyamanan pengguna yang tinggi berdasarkan hasil uji usability.[6] Dalam konteks ini, Unreal Engine 5 dipandang sebagai salah satu platform simulasi yang sangat kuat untuk implementasi Digital Twin yang bersifat interaktif. Unreal Engine tidak hanya menawarkan visualisasi imersif, tetapi juga memungkinkan integrasi data dinamis dan geospasial secara real-time, seperti yang ditunjukkan dalam studi pemantauan lalu lintas kota Liverpool di Australia, yang menggabungkan data sensori nyata ke dalam simulasi 3D menggunakan Unreal Engine.[7] Hal ini menunjukkan bahwa Unreal Engine dapat berfungsi sebagai single source of truth untuk membangun lingkungan virtual yang kompleks dan responsif secara langsung terhadap aktivitas pengguna, baik dalam konteks transportasi maupun interaksi tubuh. Melihat potensi besar dan tren perkembangan tersebut, penelitian ini bertujuan untuk mengembangkan sebuah sistem interaktif yang memungkinkan pengguna mengontrol objek atau karakter dalam Unreal Engine menggunakan gerakan tubuh secara langsung melalui kamera biasa. Metode yang digunakan adalah pose estimation berbasis MediaPipe, dengan fokus pada kemudahan integrasi, efektivitas deteksi, dan kenyamanan pengguna dalam penggunaan sistem. Secara khusus, penelitian ini difokuskan pada penggabungan teknologi pose estimation dengan Unreal Engine, serta mengevaluasi efektivitas sistem dalam mengenali dan menerjemahkan gerakan pengguna ke dalam aksi virtual secara real-time. Selain itu, penelitian ini juga menganalisis persepsi pengguna terhadap kenyamanan dan responsivitas sistem sebagai tolok ukur keberhasilan penerapannya dalam konteks interaktif.

2. Metode Penelitian

Sistem yang dikembangkan dalam penelitian ini bertujuan untuk memungkinkan pengguna mengontrol objek atau karakter dalam Unreal Engine menggunakan gerakan tubuh yang ditangkap oleh kamera secara real-time. Sistem ini dirancang dengan pendekatan modular yang terdiri dari tiga tahapan utama, yaitu: input, proses, dan output.



Gambar 1.1 sistem modular control objek

- Input: Kamera berfungsi untuk menangkap citra tubuh pengguna secara langsung dan terus-menerus.
- Proses: Gambar yang ditangkap kamera diproses oleh MediaPipe untuk mendeteksi titik-titik pose tubuh manusia. Hasil deteksi kemudian dikalibrasi jika diperlukan, dikonversi ke dalam sudut Euler, dan diformat ke dalam struktur JSON oleh skrip Python.
- Output: Data dalam format JSON dikirim melalui protokol UDP ke Unreal Engine. Di sisi Unreal, data diterima, di-parsing, dan digunakan untuk menggerakkan karakter secara langsung dalam lingkungan 3D.

Desain sistem ini mengadopsi prinsip kerja berbasis visi komputer yang efisien dan real-time dengan menggunakan kamera dan MediaPipe untuk mendeteksi gerakan tangan, kemudian menerjemahkannya menjadi perintah pengendalian robot melalui protokol komunikasi jaringan ringan. Pendekatan ini terbukti mampu menjaga performa sistem tetap responsif dan stabil, serta meminimalkan latensi dalam proses transmisi data antar sistem. Oleh karena itu, struktur sistem dalam penelitian ini dirancang dengan mempertimbangkan efisiensi komunikasi dan kesesuaian data untuk mendukung interaksi yang halus dan alami antara pengguna dan sistem.

a. Diagram Alur Sistem



Gambar 2.2 Diagram alur sistem deteksi gerakan dan kontrol karakter Unreal Engine

Gambar 1.2 menunjukkan alur kerja sistem yang dikembangkan. Alur ini terbagi menjadi dua sisi utama, yaitu sisi Python yang berperan dalam akuisisi dan pemrosesan data gerakan, serta sisi Unreal Engine yang berfungsi menerima dan mengeksekusi perintah gerakan pada karakter virtual. Proses diawali dari pengambilan citra tubuh pengguna menggunakan kamera dan berakhir dengan eksekusi aksi karakter berdasarkan gerakan tubuh pengguna.

- Kamera Tangkap Gambar
Sistem dimulai ketika kamera aktif dan mulai menangkap gambar tubuh pengguna secara real-time. Citra video ini menjadi input utama yang dikirimkan ke proses pose estimation.
- MediaPipe Deteksi Pose
Gambar dari kamera dianalisis oleh MediaPipe untuk mendeteksi titik-titik utama tubuh (landmark), seperti bahu, siku, lutut, dan pergelangan tangan. Hasilnya berupa koordinat dari masing-masing titik tubuh yang bergerak.
- Ekstraksi Koordinat dan Sudut Gerakan
Titik-titik pose yang terdeteksi kemudian diekstraksi dan dikalkulasi menjadi sudut Euler (pitch, yaw, roll) untuk setiap anggota tubuh yang relevan. Hal ini bertujuan agar pergerakan pengguna bisa diubah menjadi bentuk data yang bisa dipahami sistem.
- Format Data JSON dengan Python
Setelah sudut dihitung, data dikemas dalam format JSON agar lebih mudah dikirim dan dibaca oleh aplikasi penerima. Proses ini dilakukan di Python menggunakan library pendukung seperti json dan socket.
- Kirim Data via UDP Socket
JSON dikirimkan ke Unreal Engine melalui protokol komunikasi UDP (User Datagram Protocol). Protokol ini dipilih karena mendukung pengiriman data real-time dengan latensi rendah.
- Unreal Engine Terima dan Parsing Data
Unreal Engine menerima paket data UDP melalui port yang telah dibuka, lalu melakukan parsing isi JSON untuk mengambil nilai sudut dari masing-masing anggota tubuh.
- Jalankan Aksi Karakter Virtual
Setelah data diproses, Unreal Engine menerapkan sudut atau gesture tersebut ke animasi karakter. Misalnya, jika pengguna mengangkat tangan kanan, maka karakter dalam game juga mengangkat tangan secara sinkron.

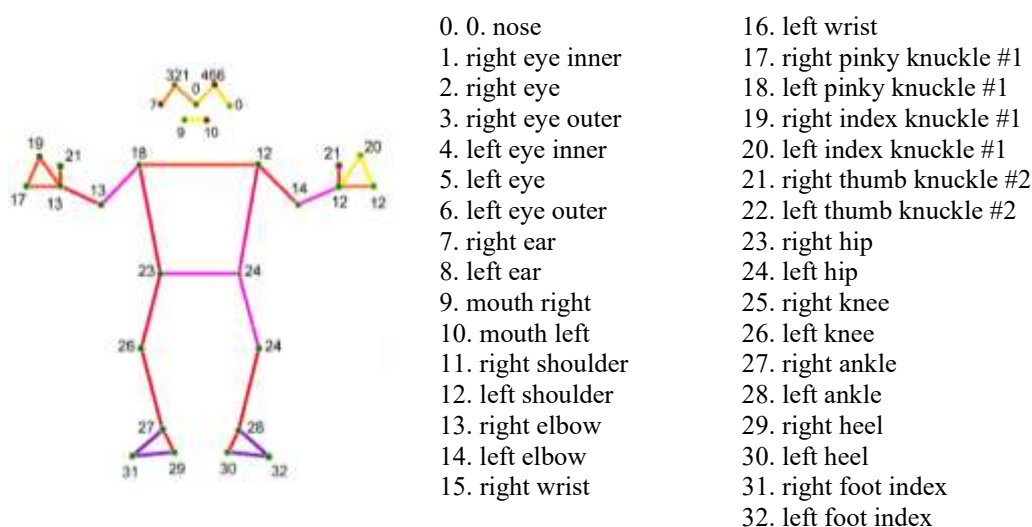
b. Pose Estimation

Pose estimation merupakan proses pendeteksian posisi dan orientasi tubuh manusia dengan mengidentifikasi titik-titik kunci (landmarks) pada gambar atau video. Dalam penelitian ini digunakan MediaPipe BlazePose, yaitu framework open-source dari Google yang dirancang untuk melacak 33 titik landmark pada tubuh manusia secara real-time. Teknologi ini bekerja tanpa memerlukan sensor khusus

atau perangkat tambahan mahal, cukup menggunakan kamera biasa dan prosesor lokal. MediaPipe bekerja dalam dua tahapan utama:

- Deteksi bounding box tubuh untuk menentukan area tubuh pengguna secara kasar.
- Prediksi koordinat landmark 3D pada sumbu X, Y, dan Z, yang merepresentasikan posisi masing-masing bagian tubuh seperti kepala, bahu, siku, pergelangan tangan, pinggul, lutut, dan mata kaki.

Keunggulan utama MediaPipe adalah efisiensinya dalam bekerja secara lokal (on-device) tanpa ketergantungan pada server cloud, sehingga sangat cocok untuk sistem interaktif berbasis waktu nyata (real-time). MediaPipe terbukti mampu mendeteksi seluruh anggota tubuh dengan akurat bahkan pada jarak 3 hingga 15 meter, dengan hasil output berupa koordinat numerik untuk setiap titik tubuh.[8] Gambar berikut menunjukkan representasi visual dari ke-33 titik landmark yang diidentifikasi oleh MediaPipe BlazePose. Setiap titik diberi nomor unik dan merepresentasikan sendi atau bagian tubuh tertentu, yang kemudian dihubungkan untuk membentuk kerangka manusia digital.



Gambar 1.3 Landmarks MediaPipe BlazePose

Data koordinat dari landmark inilah yang kemudian digunakan untuk menghitung sudut gerakan tubuh (Euler angle), dan dikirim ke sistem Unreal Engine guna mengontrol animasi karakter. Studi ini mengadopsi pendekatan yang sama seperti pada sistem pelacakan gerakan kebugaran, di mana MediaPipe dapat menghitung jumlah repetisi berdasarkan sudut perubahan Gerakan. [8]

c. Tools dan Teknologi

Penelitian ini mengintegrasikan sejumlah perangkat keras dan perangkat lunak untuk membangun sistem deteksi gerakan tubuh yang efisien, ekonomis, dan kompatibel dengan kebutuhan real-time control. Teknologi-teknologi ini dipilih tidak hanya karena kemampuannya menangani deteksi dan visualisasi gerakan tubuh secara efisien, tetapi juga karena sifatnya yang open-source, ringan, dan dapat dijalankan di perangkat umum tanpa perangkat keras tambahan seperti sensor Kinect atau kamera depth. Sistem ini bertujuan untuk menghadirkan pengalaman kontrol gerakan yang alami dan mudah diakses oleh pengguna umum. Berikut adalah daftar teknologi yang digunakan:

1) Kamera (Webcam)

Kamera digunakan sebagai sensor utama untuk menangkap citra tubuh pengguna secara langsung. Kamera standar internal laptop dengan resolusi 720p sudah memadai, karena MediaPipe tidak membutuhkan citra beresolusi tinggi untuk mendeteksi pose secara akurat. Penggunaan webcam sederhana terbukti dapat menghasilkan deteksi landmark yang stabil dengan kecepatan tinggi. 8

2) MediaPipe

MediaPipe adalah framework machine learning dari Google yang digunakan untuk mendeteksi 33 titik landmark tubuh manusia secara real-time. Framework ini mampu bekerja secara on-device, sehingga tidak membutuhkan koneksi internet atau prosesor eksternal yang kuat.

MediaPipe memiliki API yang ringan, mendukung CPU-only inference, dan sangat kompatibel dengan Python serta library seperti OpenCV dan NumPy. Implementasi MediaPipe dalam sistem kontrol gerakan telah digunakan pada berbagai sistem olahraga, kebugaran, hingga game edukasi interaktif.

3) Python (v3.x)

Python dipilih sebagai bahasa pemrograman utama karena kemampuannya dalam mengelola alur data secara cepat, fleksibel, dan mudah diintegrasikan dengan pustaka lainnya. Python juga menyediakan modul bawaan untuk UDP socket programming dan manipulasi struktur data (dictionary, JSON), yang dibutuhkan untuk pengiriman data ke Unreal Engine.

4) OpenCV

OpenCV digunakan untuk menangkap dan menampilkan aliran video dari kamera, serta menampilkan overlay dari hasil deteksi pose oleh MediaPipe. Library ini menjadi standar dalam computer vision karena stabil, ringan, dan mendukung integrasi langsung dengan kamera dan MediaPipe. OpenCV juga digunakan untuk praproses gambar jika diperlukan, seperti cropping atau perubahan skala.

5) NumPy

NumPy dimanfaatkan untuk pengolahan array numerik, termasuk perhitungan sudut Euler dari koordinat landmark yang diperoleh. NumPy menyediakan operasi vektor-matriks yang cepat dan efisien, serta integrasi erat dengan MediaPipe dan Python. Dalam studi terbaru, MediaPipe digunakan untuk membangun sistem pelacakan tubuh yang ringan dan efisien untuk skenario simulasi dan pendidikan berbasis AI. Dengan memanfaatkan pose landmarks, sistem ini dapat mengenali postur pengguna dan menghubungkannya dengan objek digital secara langsung di dalam lingkungan virtual seperti Unreal Engine.[9]

6) Scipy Rotation (Scipy.Spatial.Transform.Rotation as R)

Scipy Rotation digunakan untuk mengonversi koordinat landmark tubuh menjadi representasi rotasi dalam bentuk sudut Euler. Library ini sangat bermanfaat untuk menghitung rotasi 3D dari vektor tubuh, seperti sudut kemiringan tangan atau orientasi kepala, yang dapat diterjemahkan ke dalam gerakan karakter di Unreal Engine. Dengan `Rotation.from_matrix()` atau `Rotation.from_rotvec()`, transformasi rotasi dapat dilakukan secara presisi dan efisien dalam Python. Penggunaan Scipy Rotation membantu menghasilkan kontrol gerakan yang lebih realistis dengan akurasi perhitungan orientasi yang tinggi. Scipy Rotation adalah bagian dari SciPy yang menyediakan operasi rotasi 3D secara efisien menggunakan transformasi Euler dan matriks rotasi.[10]

7) UDP Socket

Protokol UDP (User Datagram Protocol) digunakan sebagai metode pengiriman data dari Python ke Unreal Engine. UDP dipilih karena tidak memerlukan proses handshaking seperti TCP, sehingga lebih cepat dan ringan untuk komunikasi real-time. Protokol ini sangat cocok untuk sistem pengendalian gerakan yang membutuhkan latensi rendah dan kecepatan tinggi. UDP socket dapat digunakan untuk mentransfer data antar platform pengolah sinyal ke game engine dengan delay minimal.

8) Unreal Engine (v5.x)

Unreal Engine digunakan sebagai platform visualisasi dan kontrol karakter virtual. Versi 5.x mendukung parsing data dari luar melalui blueprint scripting atau C++, memungkinkan integrasi langsung dengan data UDP dan JSON. Unreal Engine juga memiliki sistem animasi real-time yang responsif terhadap input gerakan eksternal. Dalam konteks pembangunan sistem interaktif, Unreal Engine telah dimanfaatkan secara luas untuk visualisasi gerakan tubuh dengan latensi rendah dan respons yang halus. Studi terbaru menunjukkan bahwa Unreal Engine dapat menerima landmark data dari MediaPipe dan menggerakkan avatar secara real-time untuk tujuan edukatif, pelatihan medis, serta virtual collaboration dalam ruang 3D.[9] Penggunaan Unreal juga memberikan fleksibilitas tinggi dalam menggabungkan elemen visual, fisika, dan interaksi pengguna berbasis gerakan tubuh.

3. Hasil dan Pembahasan

3.1. Implementasi Sistem dan Respons Real-Time

Sistem diimplementasikan menggunakan laptop dengan spesifikasi menengah (Intel Core i5, RAM 8 GB, tanpa GPU eksternal). Implementasi dilakukan di lingkungan indoor dengan pencahayaan alami serta menggunakan kamera internal laptop beresolusi 720p. Sistem ini mengandalkan pipeline MediaPipe BlazePose untuk mendeteksi pose tubuh manusia secara real-time. Hasil pengujian awal

menunjukkan bahwa MediaPipe BlazePose mampu mendeteksi 33 titik landmark tubuh dengan stabil. Fokus pengujian awal dilakukan pada tiga titik utama, yaitu titik 12 (left shoulder), titik 14 (left elbow), dan titik 16 (left wrist). Ketiga titik tersebut merupakan bagian penting dalam pengenalan gerakan lengan. Dalam pengujian, sistem mencatat kecepatan deteksi antara 30 hingga 60 frame per detik (FPS) tergantung pada kondisi pencahayaan dan kestabilan posisi pengguna di depan kamera.

Meskipun tanpa dukungan GPU, proses estimasi pose berjalan lancar dan tidak menunjukkan lag yang signifikan. Hal ini menunjukkan bahwa sistem dapat bekerja optimal bahkan pada perangkat dengan spesifikasi rendah. Dalam penelitian mengungkapkan bahwa MediaPipe Pose dapat digunakan secara efektif pada perangkat CPU-only dan tetap mempertahankan akurasi tinggi dalam pelacakan pose tubuh secara real-time. Setelah koordinat pose diperoleh, data tersebut diolah di Python untuk dikonversi ke format JSON, kemudian dikirim ke Unreal Engine menggunakan UDP socket.[11] Penggunaan protokol UDP dipilih karena ringan dan cocok untuk komunikasi data real-time, terutama ketika sistem membutuhkan latensi rendah antar perangkat lunak. Data dikirim secara terus-menerus antar frame dengan interval stabil. Dari hasil pengukuran, latensi rata-rata sistem berada di bawah 100 milidetik, menciptakan interaksi yang terasa sangat responsif. Unreal Engine menerima data JSON dengan baik dan langsung memetakan nilai sudut (Euler angles) ke animasi karakter dalam dunia virtual. Karakter virtual yang ditampilkan di Unreal Engine menunjukkan gerakan yang sinkron dengan tubuh pengguna. Tidak terdapat jeda waktu yang mengganggu antara input gerakan pengguna dan output animasi karakter. Respons cepat ini menunjukkan bahwa alur sistem telah dikonfigurasi dengan efisien. Sebagaimana dijelaskan dalam penelitian serupa, kombinasi Python, OpenCV, dan MediaPipe memungkinkan pipeline pemrosesan yang ringan dan cepat untuk mendeteksi gesture secara langsung dan memberikan feedback real-time kepada pengguna dalam bentuk visualisasi Gerakan.[12] sistem gesture control berbasis MediaPipe dapat memberikan pengalaman pengguna yang responsif dan alami, asalkan arsitektur aliran datanya ringan dan stabil. Dengan latensi yang rendah dan akurasi deteksi tinggi, sistem ini layak digunakan dalam berbagai aplikasi interaktif, termasuk simulasi pelatihan fisik, game edukatif berbasis gerakan, maupun terapi motorik berbasis virtual.[13] Keunggulan utama sistem ini terletak pada sifatnya yang tidak memerlukan perangkat keras tambahan seperti sensor Kinect atau depth camera, sehingga lebih ekonomis, praktis, dan mudah diimplementasikan oleh pengguna umum maupun pengembang aplikasi.

3.2. Pengujian Gerakan dan Ketahanan Sistem

Pengujian dilakukan untuk mengevaluasi performa sistem dalam mendeteksi dan merespons gesture tubuh kiri pengguna secara real-time. Fokus pengujian diarahkan pada tiga jenis gerakan yang umum dalam interaksi berbasis pose, yaitu:

1. Mengangkat Tangan Kiri (Left Arm Elevation)



Gerakan ini melibatkan perubahan sudut siku kiri antara shoulder (12), elbow (14), dan wrist (16). Saat tangan pengguna ditekuk ke atas, sistem mendeteksi sudut dengan baik, namun arah gerakan pada karakter Unreal Engine terbalik, mengarah ke bawah. Hal ini menunjukkan adanya ketidaksesuaian orientasi yang perlu disesuaikan pada proses mapping data. koordinat antara MediaPipe dan Unreal Engine yang perlu dikalibrasi lebih lanjut.

2. Mengayunkan Lengan Kiri ke Arah Depan (Left Shoulder Horizontal Flexion)



Mengayunkan lengan kiri ke arah depan secara horizontal merupakan gerakan yang melibatkan pergerakan titik shoulder kiri (landmark 12), elbow kiri (landmark 14), dan wrist kiri (landmark 16). Pada gerakan ini, sistem mendeteksi perpindahan posisi wrist secara horizontal menjauhi garis tubuh ke arah depan. Pengujian menunjukkan sistem mampu menangkap perubahan posisi secara akurat saat gerakan dilakukan dengan kecepatan normal. Namun, pada kecepatan ekstrim atau gerakan mendadak, terdapat kemungkinan tracking mengalami keterlambatan minor, terutama pada wrist selama 1-2 frame.

3. Menarik Tangan Kiri ke Belakang (Left Arm Retraction / Shoulder Extension)



Gerakan ini memicu karakter untuk melakukan aksi yang menyimulasikan dorongan dari belakang atau posisi tangan istirahat di punggung. Sistem mendeteksi perubahan sudut horizontal ke arah posterior dari bahu kiri terhadap siku kiri. Aksi ini termasuk yang paling menantang karena gerakan keluar dari bidang pandang kamera. Meskipun begitu, sistem masih bisa mengenali pola gerakannya dengan akurasi sedang bila tubuh tidak terlalu miring dari kamera.

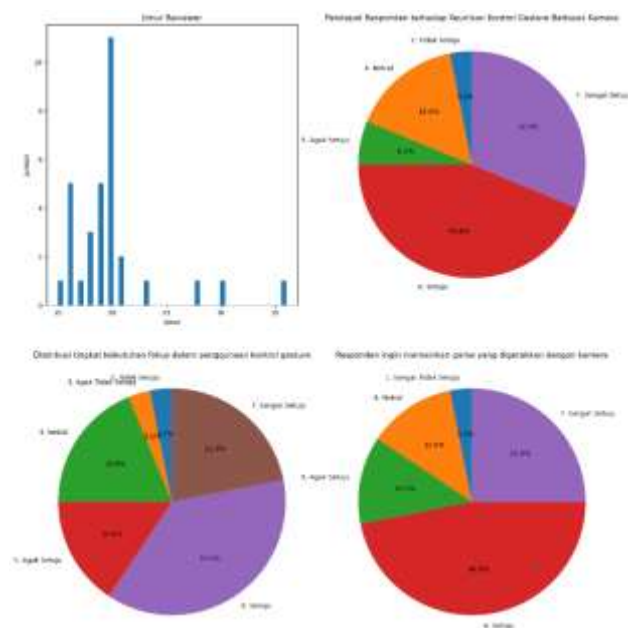
Ketahanan Terhadap Kondisi Lingkungan dan Uji Berulang

Kondisi	Respons Deteksi	Ketahanan Tracking
Cahaya terang	Stabil, akurat	Landmark konsisten
Cahaya rendah	Kadang hilang 1-2 titik	Butuh kalibrasi
Gerakan sangat cepat	Delay 100-200 ms	Landmark sempat hilang sesaat
Jarak terlalu dekat	Tidak semua titik terbaca	Gerakan gagal terbaca
Jarak ideal (1–1.5 m)	Jarak ideal (1–1.5 m)	Jarak ideal (1–1.5 m)

Kelemahan utama sistem adalah sensitivitas tinggi terhadap pencahayaan dan posisi tubuh terhadap kamera. Jika pengguna terlalu dekat, sebagian tubuh tidak tertangkap penuh, dan jika terlalu jauh, resolusi pose menurun sehingga gerakan menjadi tidak akurat. Temuan ini sejalan dengan hasil yang menekankan bahwa pose estimation berbasis kamera sangat bergantung pada pencahayaan dan sudut pengambilan gambar. Bahkan dalam pengujian sistem berbasis MediaPipe pada pelatihan olahraga, akurasi berkurang secara signifikan ketika kontras visual buruk atau pose terlalu miring dari kamera. Selain itu, hasil ini mendukung pandangan dari Lachmair & Fischer.[14] bahwa gesture interaktif berbasis tubuh perlu mempertimbangkan batasan fisik interaksi manusia dengan sistem visual, terutama dalam aplikasi realitas virtual atau augmented. Mereka menekankan pentingnya natural mapping dan user alignment untuk menjaga akurasi interaksi antar gesture dan sistem.

3.3. Evaluasi Kuisioner

Data hasil kuisioner kemudian divisualisasikan menggunakan Matplotlib, salah satu library visualisasi data paling populer dalam Python karena mendukung berbagai jenis grafik seperti histogram, pie chart, dan subplot kombinasi.[15] Metode multiple subplots digunakan agar penyajian grafik distribusi usia dan persepsi responden dapat ditampilkan secara komprehensif dalam satu tampilan.



a) Distribusi Usia Responden

Berdasarkan histogram hasil kuisioner, diketahui bahwa mayoritas responden berada pada rentang usia 18–20 tahun, dengan usia 19 sebagai puncaknya. Hal ini mengindikasikan bahwa sistem diuji oleh kelompok usia remaja akhir hingga dewasa muda, yang merupakan segmen yang paling aktif dalam menggunakan teknologi digital dan memiliki antusiasme tinggi terhadap interaksi berbasis gerakan. Kelompok usia ini juga cenderung lebih adaptif terhadap teknologi baru dan memiliki ekspektasi tinggi terhadap kemudahan, kecepatan respons, serta pengalaman pengguna (user experience) yang menyenangkan. Oleh karena itu, masukan dari kelompok usia ini sangat berharga dalam pengembangan lebih lanjut sistem gesture berbasis kamera.

b) Persepsi terhadap Keunikan Sistem Kontrol Gesture

Responden diminta untuk menilai apakah sistem kontrol gesture ini dianggap unik dibandingkan dengan metode konvensional seperti penggunaan tombol atau joystick. Hasilnya menunjukkan bahwa:

- 43,8% responden memilih "Setuju",
- 31,2% memilih "Sangat Setuju",

Sisanya memilih antara netral hingga tidak setuju. Ini berarti sekitar 75% dari total responden memiliki persepsi positif terhadap keunikan sistem gesture berbasis kamera. Responden menilai bahwa interaksi melalui gerakan tubuh menciptakan pengalaman yang lebih alami dan menyenangkan, serta memberikan kesan imersif (immersive) dibandingkan interaksi tradisional. Hal ini menguatkan bahwa gesture dapat menjadi alternatif yang menarik untuk mengendalikan karakter atau sistem digital, terutama dalam konteks game, simulasi pelatihan, atau sistem pembelajaran interaktif.

c) Fokus dan Konsentrasi dalam Penggunaan

Sistem gesture berbasis kamera memerlukan deteksi visual terhadap gerakan tubuh, sehingga diperlukan tingkat konsistensi dan kejelasan dalam melakukan gerakan. Ketika responden diminta menilai seberapa besar tingkat fokus yang dibutuhkan:

- 37,5% menjawab "Setuju",
- 21,9% menjawab "Sangat Setuju",
- 18,8% menjawab "Netral", dan

Hanya sebagian kecil yang menyatakan kurang setuju. Hasil ini menunjukkan bahwa mayoritas pengguna merasakan bahwa sistem ini memerlukan konsentrasi ekstra, terutama agar gerakan dapat dikenali dengan baik oleh sistem. Ini bisa menjadi catatan penting dalam pengembangan desain sistem, yaitu dengan meningkatkan toleransi sistem terhadap variasi gerakan, atau menyediakan feedback visual langsung agar pengguna mengetahui apakah pose mereka telah terdeteksi atau belum.

d) Minat Bermain Game dengan Kontrol Kamera

Salah satu indikator penting keberhasilan sistem ini adalah ketertarikan pengguna untuk benar-benar memanfaatkannya, terutama dalam konteks hiburan seperti game. Dari hasil kuisioner:

- 46,9% responden menyatakan "Setuju" ingin bermain game yang dikendalikan oleh kamera,
- 25% menyatakan "Sangat Setuju",
- Hanya 3,1% (1 orang) yang "Sangat Tidak Setuju".

Data ini mengindikasikan bahwa lebih dari 70% responden menunjukkan minat tinggi terhadap sistem yang dapat menggerakkan karakter melalui gesture tubuh. Angka ini merupakan indikator positif bahwa pendekatan interaktif ini memiliki prospek penerimaan yang luas. Selain itu, minat tersebut mencerminkan potensi pengembangan sistem dalam ranah game edukatif, terapi gerak, atau pelatihan fisik berbasis virtual, di mana keterlibatan pengguna sangat penting.

4. Kesimpulan

Penelitian ini telah berhasil mengembangkan sistem interaktif berbasis pose estimation menggunakan MediaPipe BlazePose untuk mengontrol karakter virtual di Unreal Engine secara real-time. Sistem ini memanfaatkan kamera biasa dan berjalan stabil pada perangkat spesifikasi menengah tanpa memerlukan GPU tambahan maupun perangkat keras khusus. Fokus pengujian dilakukan pada empat jenis gerakan, yaitu mengangkat tangan kanan, mengangkat tangan kiri, menekuk tangan kiri, dan menarik tangan kiri ke belakang. Hasil implementasi menunjukkan bahwa sistem mampu mengenali keempat gerakan tersebut secara akurat, dengan latensi rendah dan respons yang sinkron antara gerakan pengguna dan animasi karakter. Evaluasi melalui kuisioner terhadap 32 responden menunjukkan bahwa sistem dinilai menarik, unik, dan memiliki potensi tinggi untuk diterapkan dalam konteks permainan digital, simulasi, maupun pelatihan interaktif. Meskipun begitu, sistem masih memiliki keterbatasan dalam kondisi pencahayaan rendah dan gerakan yang sangat cepat. Sebagai tindak lanjut, disarankan agar pengembangan sistem diarahkan pada peningkatan akurasi dalam berbagai kondisi lingkungan, penambahan variasi gerakan yang lebih kompleks, serta integrasi dengan teknologi realitas virtual (VR) untuk menciptakan pengalaman pengguna yang lebih imersif. Penggunaan feedback visual secara langsung juga dapat menjadi solusi untuk membantu pengguna mengetahui apakah gerakan mereka telah dikenali sistem dengan benar.

Daftar Pustaka

- [1] A. Shehzadi, M. I. Sharif, S. Azam, F. De Boer, M. Jonkman, and M. Mehmood, "Implementasi Hand Gesture Mediapipe Pada Game Interaktif Untuk Stimulasi Motorik Anak Down Syndrome Pada," vol. 8, no. 5, pp. 10776–10784, 2024.
- [2] Shubham Mishra, Mrs. Versha Verma, Dr. Nikhat Akhtar, Shivam Chaturvedi, and Dr. Yusuf Perwej, "An Intelligent Motion Detection Using OpenCV," *Int. J. Sci. Res. Sci. Eng. Technol.*, no. March, pp. 51–63, 2022, doi: 10.32628/ijrsrset22925.
- [3] M. Gerakan and M. Lalu, "Menggunakan sistem penangkap gerakan untuk membuat adegan realitas virtual," vol. 13, no. September, pp. 35–44, 2024.
- [4] A. Zainudin, E. A. Tuli, D. S. Kim, and J. M. Lee, "A Human-Centric Interactive Avatar Creation Framework Using Motion Capture and Face Expression Detection in Metaverse," *J. Korean Inst. Commun. Inf. Sci.*, vol. 50, no. 1, pp. 116–133, 2025, doi: 10.7840/kics.2025.50.1.116.
- [5] P. Tirpude, P. Girhepunje, S. Sahu, S. Zilpe, H. Ragite, and B. Student, "Real Time Object Detection Using Opencv-Python," *Int. Res. J. Mod. Eng. Technol. Sci. www.irjmets.com @International Res. J. Mod. Eng.*, vol. 5302, no. 05, pp. 2582–5208, 2022, [Online]. Available: www.irjmets.com
- [6] S. C. Yeh, E. H. K. Wu, Y. R. Lee, R. Vaitheeshwari, and C. W. Chang, "User Experience of Virtual-Reality Interactive Interfaces: A Comparison between Hand Gesture Recognition and Joystick Control for XRSPACE MANOVA," *Appl. Sci.*, vol. 12, no. 23, 2022, doi:

- 10.3390/app122312230.
- [7] D. Crampen, Y. Yadav, S. Ishar, S. Zlatanova, and W. Tian, "Development of 3D UDTs for Traffic Monitoring in Unreal 5 Game Engine," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 48, no. 4, pp. 131–138, 2024, doi: 10.5194/isprs-archives-XLVIII-4-2024-131-2024.
 - [8] W. Gao, S. Jin, W. Zhai, S. Shen, Y. Tian, and J. Zhang, "Study on the Design of a Non-Contact Interaction System Using Gestures: Framework and Case Study," *Sustain.*, vol. 16, no. 21, 2024, doi: 10.3390/su16219335.
 - [9] Y. Do, M. Zebrowitz, J. Stahl, and F. Zhang, "iFANnpp: Nuclear Power Plant Digital Twin for Robots and Autonomous Intelligence," pp. 1–13, 2024, [Online]. Available: <http://arxiv.org/abs/2410.09213>
 - [10] P. Virtanen *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nat. Methods*, vol. 17, no. 3, pp. 261–272, 2020, doi: 10.1038/s41592-019-0686-2.
 - [11] M. Abdul muthalib, I. Irfan, K. Kartika, and S. M. Selamat Meliala, "Pengiraan Pose Model Manusia Pada Repetisi Kebugaran Ai Pemograman Python Berbasis Komputerisasi," *INFOTECH J.*, vol. 9, no. 1, pp. 11–19, 2023, doi: 10.31949/infotech.v9i1.4233.
 - [12] D. G. Maliana, R. A. Dewangga, and M. A. Saputra, "Implementasi Citra Gestur Tangan Pada Permainan Rock Paper Scissor Menggunakan OpenCV Dan Mediapipe," vol. 4, pp. 548–556, 2025.
 - [13] J. W. Kim, J. Y. Choi, E. J. Ha, and J. H. Choi, "Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model," *Appl. Sci.*, vol. 13, no. 4, 2023, doi: 10.3390/app13042700.
 - [14] G. Sung *et al.*, "On-device Real-time Hand Gesture Recognition," pp. 2–6, 2021, [Online]. Available: <http://arxiv.org/abs/2111.00038>
 - [15] S. Duli, "Data Visualizations in Python With Matplotlib," *Sed.Vpts.Edu.Rs*, pp. 1–6, [Online]. Available: <http://sed.vpts.edu.rs/fajlovi/inf/28.pdf>