

Implementasi Moving Average dalam Pergerakan Objek 3D Dimensi Berdasarkan Gerakan Tangan Menggunakan Mediapipe

Fachri Yanuar Rudi F^{*1}, Guntur Syahputra², Safriadi³

^{1,2,3}Jurusan Teknologi Informasi dan Komputer, Politeknik Negeri Lhokseumawe

Email: ¹fachri@pnl.ac.id, ²guntur@pnl.ac.id, ³safriadi@pnl.ac.id

*Penulis Korespondensi

Abstrak

Perkembangan teknologi computer vision memungkinkan interaksi manusia-komputer melalui deteksi gerakan tubuh secara real-time. Salah satu solusi yang banyak digunakan adalah MediaPipe, yang mampu melakukan pelacakan gerakan tangan tanpa memerlukan sensor tambahan. Namun, hasil akuisisi gerakan tangan dengan MediaPipe sering mengalami *jitter*, yaitu fluktuasi kecil yang mengganggu kestabilan data, terutama saat data tersebut diimplementasikan pada objek 3D. Penelitian ini bertujuan untuk mengurangi *jitter* pada hasil akuisisi gerakan tangan menggunakan MediaPipe dengan menerapkan metode penyaringan data berupa moving average filter. Data posisi dan rotasi tangan yang diperoleh dari MediaPipe dianalisis dan difilter, kemudian hasilnya digunakan untuk menggerakkan objek 3D secara lebih halus. Hasil penelitian menunjukkan bahwa *moving average* filter cukup efektif dalam mengurangi *jitter* pada data posisi tangan. Namun, pada data rotasi tangan, filter ini kurang memberikan hasil optimal karena nilai rotasi cenderung lebih fluktuatif. Oleh karena itu, diperlukan penelitian lanjutan dengan metode penyaringan sinyal untuk mengatasi jitter pada rotasi. Temuan ini menunjukkan potensi awal untuk mengintegrasikan pelacakan tangan berbasis MediaPipe dengan objek 3D secara lebih stabil dan responsif.

Kata kunci: Jitter, MediaPipe, Moving Average Filter, Pelacakan Gerakan Tangan

Abstract

The advancement of computer vision technology has enabled real-time human-computer interaction through body motion tracking. One widely used solution is MediaPipe, which can track hand movements without requiring additional sensors. However, hand motion acquisition results using MediaPipe often suffer from jitter—small, rapid fluctuations that reduce data stability—especially when applied to 3D objects. This study aims to reduce jitter in hand movement data acquired through MediaPipe by applying a data smoothing technique using the moving average filter. Both hand position and rotation data obtained from MediaPipe were filtered and then used to drive a 3D object with smoother movement. The results show that the moving average filter is effective in reducing jitter in hand position data. However, it is less effective for hand rotation data due to the highly fluctuating nature of rotational values. Therefore, further research using alternative filtering methods is needed to better address jitter in hand rotation. These findings provide an initial step toward integrating MediaPipe-based hand tracking with 3D objects in a more stable and responsive manner.

Keywords: Hand Motion Tracking, Jitter, MediaPipe, Moving Average Filter

I. PENDAHULUAN

Dalam beberapa tahun terakhir, teknologi pelacakan gerakan tangan semakin banyak dimanfaatkan dalam berbagai aplikasi, mulai dari augmented reality (AR), virtual reality (VR), hingga pengendalian objek digital secara langsung menggunakan gestur. Salah satu teknologi awal yang cukup populer adalah Microsoft Kinect, yang mampu mendeteksi gerakan tubuh dan tangan menggunakan sensor kedalaman (depth sensor). Melalui teknologi ini, peneliti seperti Ren dan rekan-rekannya (2011) berhasil mengembangkan sistem pengenalan gestur tangan yang mampu mengenali bentuk tangan secara cukup akurat bahkan dalam kondisi pencahayaan yang kompleks [1]. Penelitian lain oleh Chang (2016) juga menunjukkan bahwa ujung jari tangan dapat dilacak secara real-time dalam ruang tiga dimensi dengan akurasi yang tinggi menggunakan Kinect[2]. Penelitian lainnya oleh Rudi F (2021) melakukan pendekatan dengan menggunakan *skeleton* tangan yang didapat dari hasil pengukuran kedalaman kamera dapat digunakan untuk tracking telapak tangan [3].

Namun, meskipun efektif, Kinect memiliki beberapa keterbatasan. Selain ukurannya yang relatif besar dan memerlukan perangkat keras khusus, harganya juga cukup tinggi dan tidak terlalu praktis untuk digunakan dalam skala kecil atau portabel.

Seiring berkembangnya teknologi, muncul pendekatan baru yang jauh lebih ringan dan fleksibel, yaitu menggunakan MediaPipe, sebuah *framework open-source* dari Google yang dapat mendeteksi dan melacak gerakan tangan secara *real-time* hanya dengan menggunakan kamera biasa. MediaPipe Hands mampu mendeteksi 21 titik kunci (*landmark*) pada tangan secara presisi dan cepat, bahkan dapat dijalankan langsung di perangkat mobile [4]. Pada penelitian sebelumnya, MediaPipe telah digunakan untuk mengenali posisi dan arah jari tangan, dan hasilnya cukup menjanjikan dalam konteks interaksi manusia dan komputer.

Penggunaan MediaPipe dalam penelitian ini bukan tanpa alasan. Selain karena tidak memerlukan perangkat tambahan seperti sensor kedalaman, MediaPipe juga terbukti mampu bekerja dengan cepat dan akurat. Menurut Zhang dan timnya (2020), MediaPipe Hands mampu menjalankan pelacakan tangan secara *real-time* dengan latensi rendah, dan sangat cocok untuk berbagai aplikasi yang membutuhkan interaksi berbasis gerakan tangan [4].

Meski demikian, sistem pelacakan berbasis kamera seperti MediaPipe tidak sepenuhnya bebas dari kendala. Salah satu masalah umum yang sering muncul adalah *jitter*, yaitu gerakan yang tampak bergetar atau tidak stabil akibat fluktuasi kecil pada posisi titik *landmark* dari *frame* ke *frame*. Jika tidak diatasi, *jitter* ini akan sangat memengaruhi kenyamanan pengguna, terutama ketika gerakan tangan digunakan untuk mengendalikan objek 3D secara langsung.

Untuk mengatasi masalah tersebut, penelitian ini menerapkan metode *moving average*, sebuah metode penyaringan sederhana namun efektif untuk menghaluskan gerakan. Dengan menghitung rata-rata dari sejumlah *frame* sebelumnya, sistem menjadi lebih stabil dan gerakan objek 3D terlihat lebih halus. Metode ini juga sangat cocok untuk aplikasi *real-time* karena ringan dan mudah diimplementasikan [5].

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengimplementasikan metode *moving average* pada pelacakan tangan menggunakan MediaPipe, agar pergerakan objek 3D yang dikendalikan oleh gerakan tangan dapat tampil lebih stabil dan responsif.

II. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental dengan menerapkan integrasi antara sistem pelacakan gestur tangan menggunakan MediaPipe dan manipulasi objek 3 dimensi di dalam Blender. Tahapan metode yang digunakan dalam penelitian ini dijabarkan sebagai berikut:

2.1. Perancangan Sistem

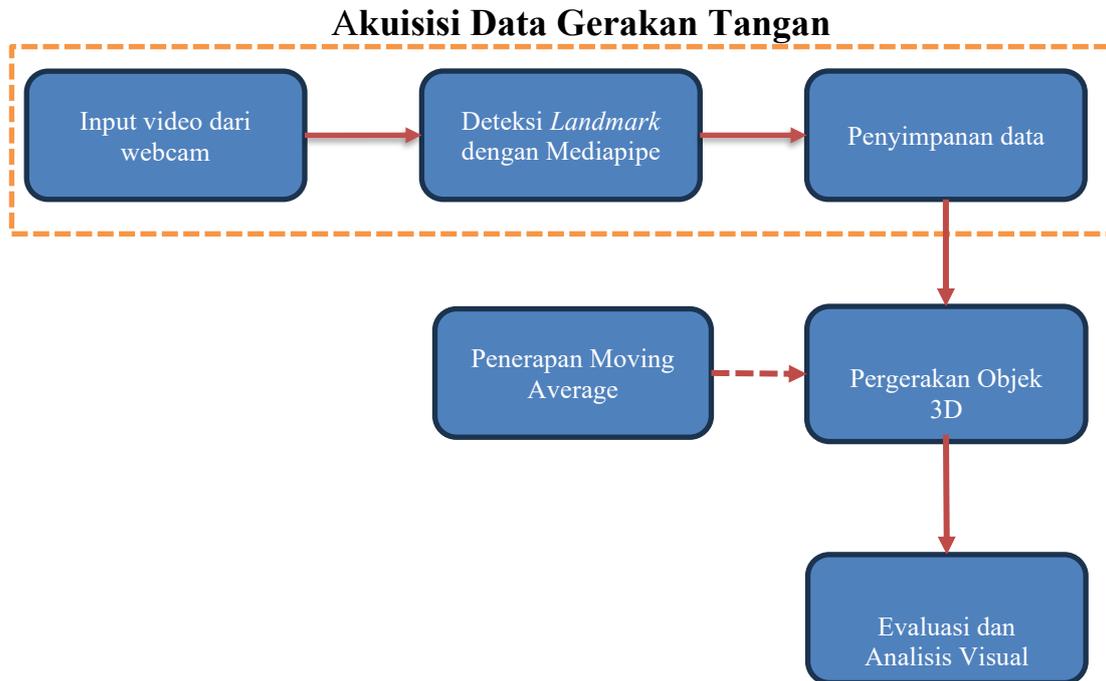
Langkah awal dalam penelitian ini adalah merancang alur sistem yang menghubungkan input gestur tangan dengan pergerakan objek 3D. Seperti yang terlihat pada gambar 1 sistem dibagi menjadi tiga komponen utama:

Akuisisi Data Gestur: Menggunakan MediaPipe untuk mendeteksi dan melacak posisi tangan dan jari secara *real-time* melalui webcam. Pada penelitian ini data dari jari yang digunakan adalah ujung jari telunjuk. Penggunaan mediapipe dipilih karena mediapipe memiliki beberapa kelebihan di antaranya adalah dapat mendeteksi tangan secara waktu nyata (*realtime*) memiliki sedikit jeda (*delay*) dan yang terpenting adalah tanpa perlu menggunakan perangkat khusus seperti microsoft Kinect atau *Hand Tracker Camera*. Mediapipe dapat mendeteksi tangan dengan hanya menggunakan webcam.

Pemrosesan data *landmark*, data koordinat dari MediaPipe diproses untuk mengidentifikasi posisi dan perputaran (rotasi) dari jari tangan yang akan dipetakan pada objek 3D. data yang diperoleh akan disimpan dalam bentuk file berupa file CSV. Pada tahap ini data yang digunakan adalah hasil pelacakan dari ujung jari telunjuk.

Penerapan filter *moving average* pada hasil data *landmark* bertujuan untuk mereduksi *jitter* yang terjadi pada saat pemrosesan data sehingga gerak dari objek 3 dimensi nantinya akan lebih halus.

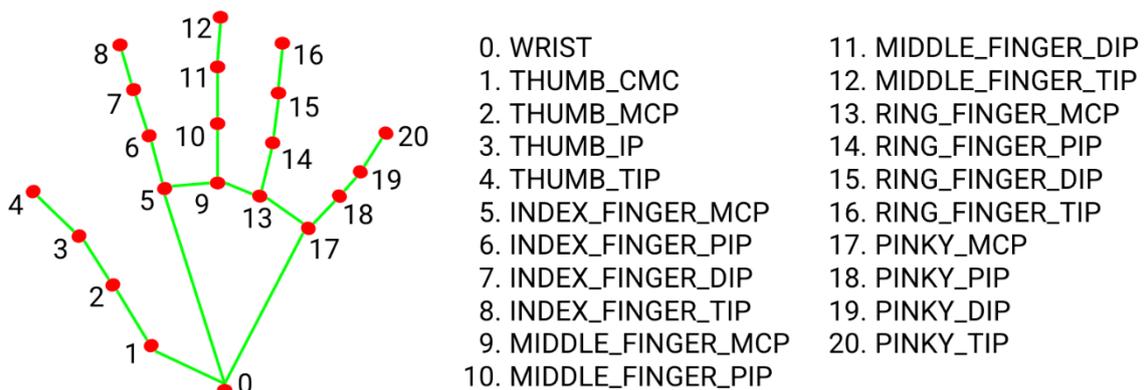
Objek 3D di Blender: Posisi atau orientasi objek dalam Blender diubah berdasarkan interpretasi dari posisi dan rotasi jari tangan.



Gambar 1. Komponen Utama Sistem

2.2. Akuisisi data gerakan tangan

MediaPipe diimplementasikan menggunakan bahasa pemrograman Python. Modul hand tracking dari MediaPipe digunakan untuk memperoleh titik-titik landmark pada tangan, pada penelitian ini data landmark yang digunakan adalah data dari ruas ujung jari telunjuk, titik-titik landmark tangan mediapipe dapat dilihat pada gambar 2. Data koordinat (x, y, z) dan juga perhitungan rotasi dari jari telunjuk tersebut akan digunakan sebagai input untuk mengontrol objek 3D dan disimpan ke dalam file CSV. File CSV tersebut akan mengandung data berupa frame, koordinat (x,y,z) dan rotasi pada sumbu x, y, dan z.



Gambar 2. Landmark tangan mediapipe

Sumber: <https://mediapipe.readthedocs.io/en/latest/solutions/hands.html>

Perhitungan rotasi ujung jari tangan dilakukan dengan mengitung koordinat ujung jari dengan nomor *landmark* 7 dan dibandingkan dengan koordinat nomor *landmark* 8 kemudian dihitung vektor dari kedua koordinat tersebut. Persamaan yang digunakan untuk menghitung rotasi ujung jari telunjuk dapat

dilihat pada persamaan [1] sampai [4].

Menghitung Vektor dari landmark 7 ke landmark 8

$$\begin{aligned}vx &= x8 - x7 \\vy &= y8 - y7 \\vz &= z8 - z7\end{aligned}\quad [1]$$

Perhitungan sudut rotasi terhadap Sumbu Z (rotasi di bidang XY).

$$\theta_z = \arctan2(vy, vx) \quad [2]$$

Perhitungan sudut rotasi terhadap Sumbu X (rotasi di bidang YZ).

$$\theta_x = \arctan2(vz, vy) \quad [3]$$

Perhitungan sudut rotasi terhadap Sumbu Y (di bidang XZ).

$$\theta_y = \arctan2(vz, vx) \quad [4]$$

Dimana vx, vy, vz adalah vektor sumbu X, sumbu Y dan sumbu Z. $x8, y8, z8$ adalah koordinat dari titik landmark 8 dan $x7, y7, z7$ adalah koordinat dari titik landmark 7. Perhitungan rotasi ini menghasilkan nilai dalam bentuk radian. Hasil dari implementasi mediapipe ini adalah titik koordinat dan rotasi dari ujung jari telunjuk. Titik koordinat dan roatsi ini kemudian akan disimpan ke dalam file dengan format CSV.

2.3. Implementasi Moving Average

Hasil dari akuisisi data gerakan tangan masih terdapat *jitter* jika langsung diimplementasikan. Untuk mengurangi *jitter* tersebut dilakukan pengurangan *jitter* dengan menggunakan metode moving average. Moving average merupakan metode yang sering digunakan untuk melakukan prediksi data pada kumpulan data berdasarkan data terdahulu [6]. Setiap koordinat dan juga hasil perhitungan rotasi dari landmark (x, y, z) difilter menggunakan rumus moving average sederhana seperti pada persamaan [5].

$$\bar{P}_t = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i} \quad [5]$$

Dimana:

\bar{P}_t : posisi rata-rata pada waktu ke-t,

P_{t-i} : posisi landmark pada frame sebelumnya,

n : ukuran jendela (window size), misalnya 5 atau 10 frame.

2.4. Integrasi dengan Blender

Blender digunakan sebagai platform visualisasi objek 3D. Dengan menggunakan Blender Python API (bpy), objek dalam scene dapat digerakkan sesuai dengan input gerakan tangan. Integrasi dilakukan melalui pembacaan file CSV yang telah dibuat dan telah difilter sebelumnya. Setelah dilakukan prapengolahan data, kemudian data tersebut akan dijadikan sebagai posisi dan rotasi dari objek 3 dimensi. Setiap frame yang ditangkap dan direkam oleh mediapipe akan diterjemahkan menjadi *keyframe* pada blender.

III. HASIL DAN PEMBAHASAN

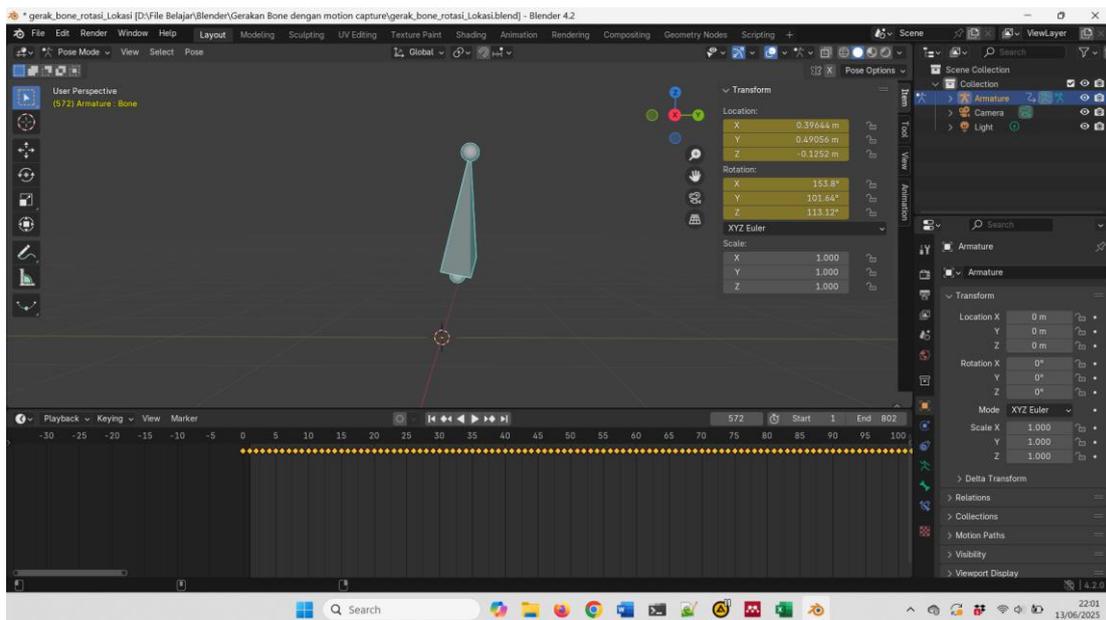
Untuk mendukung proses analisis, langkah awal yang dilakukan adalah melakukan akuisisi data gerakan tangan menggunakan MediaPipe. Proses ini menghasilkan kumpulan data berupa koordinat tiga

dimensi (x, y, z) dari 21 titik landmark pada tangan, yang dideteksi secara real-time dari tangkapan kamera. Gambar 3 memperlihatkan tangkapan layar saat sistem berhasil mendeteksi kerangka tangan dengan baik, di mana titik-titik pada jari dapat dilacak secara konsisten dari frame ke frame.



Gambar 3. Hasil deteksi *landmark* tangan, lokasi dan juga perhitungan rotasi dari ujung jari telunjuk

Selanjutnya, data hasil akuisisi tersebut digunakan sebagai dasar untuk menggerakkan objek 3D dalam Blender. Implementasi ini dilakukan melalui dua tahap: pertama, memetakan gerakan jari (terutama ujung jari telunjuk) ke dalam ruang kerja 3D di Blender; dan kedua, mengatur pergerakan objek melalui keyframe yang dihasilkan dari koordinat tangan tersebut. Pada gambar 4 dapat dilihat bahwa data hasil akuisisi gerakan tangan berhasil diimplementasikan pada Blender. Objek 3D bergerak dan berotasi sesuai dengan gerakan tangan. Data hasil gerakan tangan tersebut diimplementasikan sebagai *keyframe* pada blender sehingga dapat menjadi sumber gerakan animasi. Data yang dimasukkan ke dalam blender sudah dilakukan penerapan filter sehingga gerakan objek 3D menjadi lebih halus dan minim *jitter*.



Gambar 4. Hasil integrasi data pergerakan tangan dengan Blender

Data pada tabel 1 merupakan potongan data dari data rotasi dan posisi jari telunjuk hasil dari akuisisi data menggunakan mediapipe, sedangkan tabel 2 menunjukkan data hasil penerapan filter pada rotasi dan posisi jari.

Tabel 1. Potongan Data Akuisisi Data Menggunakan Mediapipe

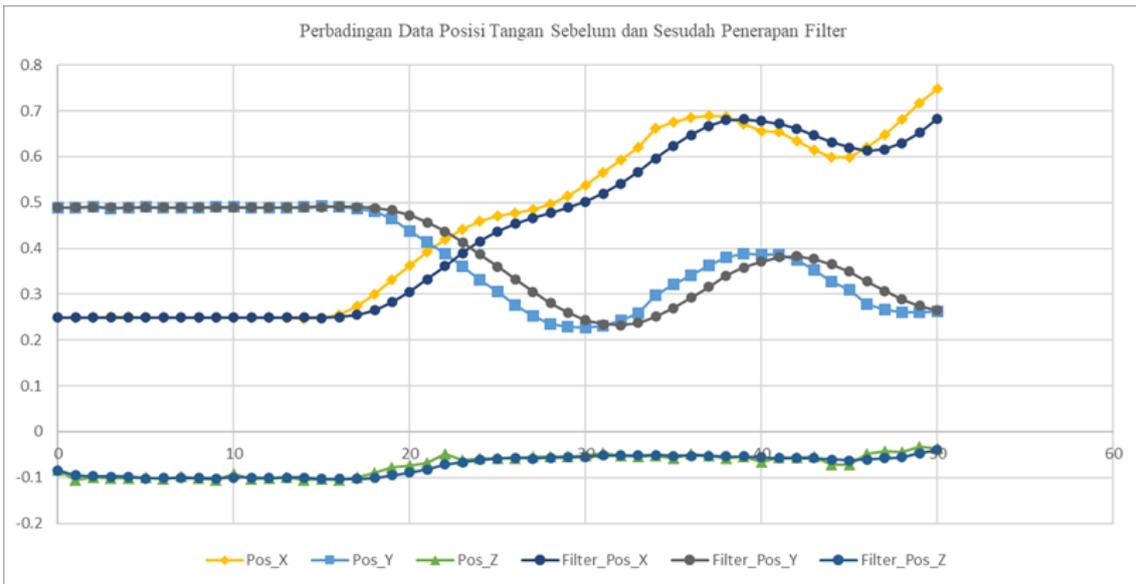
Frame	Rotasi X	Rotasi Y	Rotasi Z	Posisi X	Posisi Y	Posisi Z
0	485.2460224	947.8306	593.1898	0.249213308	0.489223272	-0.084016383
1	490.5689137	945.0779	598.0115	0.249351516	0.489378363	-0.106298298
2	489.7343003	948.9636	59.36679	0.248690695	0.4910779	-0.099898852
3	490.2797286	947.3925	595.4997	0.250188768	0.487072825	-0.101713359
4	487.9697337	947.0087	595.1059	0.250538737	0.488390297	-0.100556485
5	491.3645883	948.3843	594.8141	0.248724803	0.490079492	-0.099549487
6	486.5365589	9.456541	59.60116	0.249437302	0.489516526	-0.103790507
7	49.18585222	95.017	59.3107	0.249245092	0.489398301	-0.097361691
8	490.5261334	947.4902	595.4782	0.248508021	0.487791777	-0.10241089
9	491.486136	947.0681	596.225	0.248699218	0.489740252	-0.10603828
10	488.8886972	95.10368	591.1817	0.249276519	0.491046816	-0.091982424

Data pada tabel hanya ditunjukkan 10 frame, sedangkan pada saat dilakukan penelitian, frame dibatasi sebanyak 100 frame. Data pada tabel 2 menunjukkan data yang telah dilakukan filter.

Tabel 2. Potongan Data yang Telah diterapkan Filter

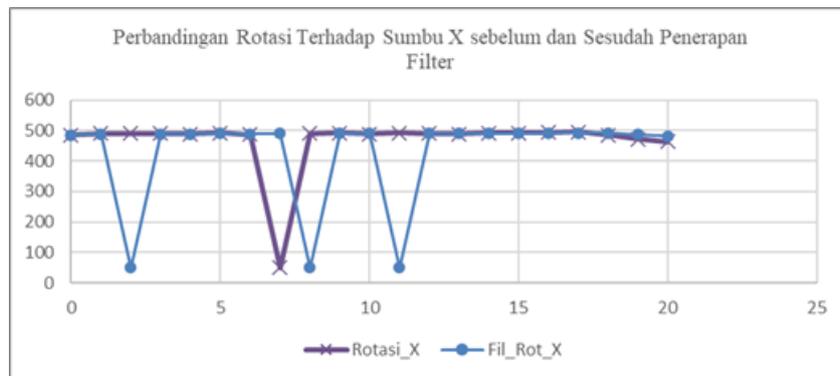
Frame	Rotasi_X	Rotasi_Y	Rotasi_Z	Filter_Pos_X	Filter_Pos_Y	Filter_Pos_Z
0	485.246	947.8306	593.1898	0.249213	0.489223	-0.08402
1	487.9075	946.4543	595.6006	0.249282	0.489301	-0.09516
2	48.85164	947.2907	594.9564	0.249085	0.489893	-0.09674
3	488.9572	947.3162	595.0922	0.249361	0.489188	-0.09798
4	488.7597	947.2547	595.0949	0.249597	0.489029	-0.0985
5	489.9835	947.3654	595.4198	0.249499	0.4892	-0.1016
6	489.177	947.4806	595.0198	0.249516	0.489227	-0.1011
7	489.6018	94.77219	594.9077	0.249627	0.488891	-0.10059
8	48.96511	94.77414	594.9034	0.249291	0.489035	-0.10073
9	490.3544	947.7533	595.1272	0.248923	0.489305	-0.10183
10	489.8592	9.482838	594.4007	0.249033	0.489499	-0.10032

Dari data pada tabel 1 dan 2 dapat dilihat adanya sedikit perubahan nilai. Data rotasi terhadap sumbu x pada tabel 1 frame 2 bernilai 490.568, setelah dilakukan penerapan filter berubah menjadi 487.907, kemudian pada frame 3 di tabel 1 posisi y bernilai 0.48707 sedangkan pada tabel 2 0.48918. perubahan itu terjadi dikarenakan pada saat dilakukan filtering, maka akan dilakukan penyesuaian data hasil filtering. Untuk lebih jelasnya data posisi jari sebelum dan sesudah filtering akan ditampilkan pada gambar 5.



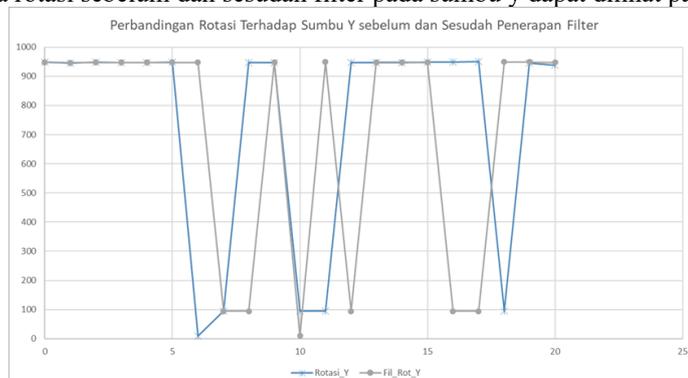
Gambar 5. Perbandingan data posisi gerakan tangan sebelum dan sesudah diterapkan filter

Dari gambar 5 dapat dilihat bahwa ada perubahan data posisi yang terjadi setelah dilakukan filter. Filter moving average yang diterapkan mengubah data awal menjadi data yang lebih halus dan terbebas dari *jitter*. Perbandingan data rotasi pada sumbu x sebelum dan sesudah penerapan filter *moving average* dapat dilihat pada gambar 6.



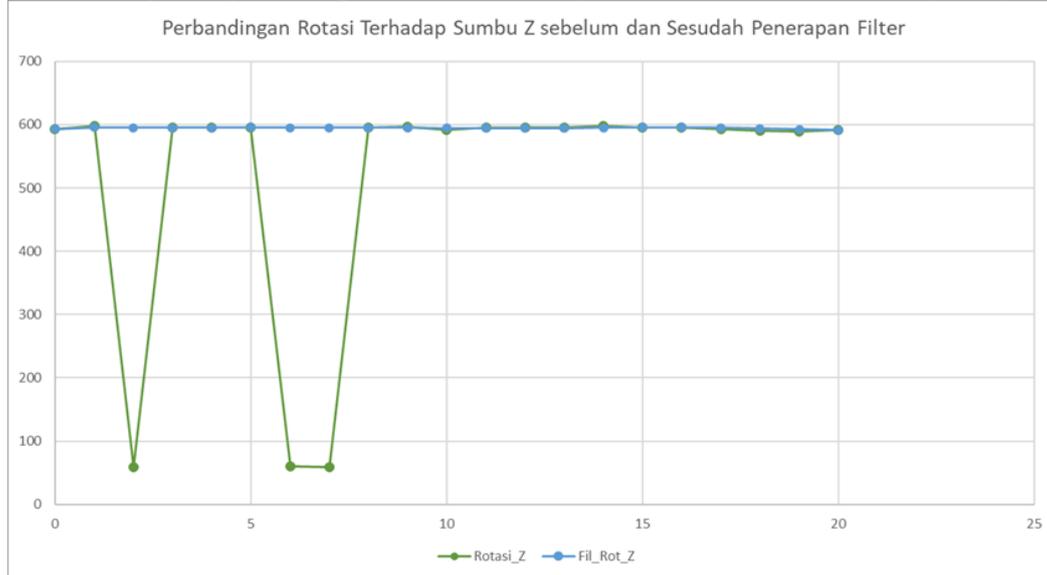
Gambar 6. Perbandingan data rotasi terhadap sumbu x sebelum dan sesudah diterapkan filter

Dari gambar 6 dapat terlihat adanya perubahan data rotasi sebelum dan sesudah penerapan filter. Untuk perbandingan data rotasi sebelum dan sesudah filter pada sumbu y dapat dilihat pada gambar 7.



Gambar 7. Perbandingan data rotasi terhadap sumbu y sebelum dan sesudah diterapkan filter

Gambar 7 juga menunjukkan adanya perubahan data dari sebelum dan sesudah penerapan filter. Gambar 8 menunjukkan perbandingan data rotasi terhadap sumbu z sebelum dan sesudah penerapan filter.



Gambar 8. Perbandingan data rotasi terhadap sumbu z sebelum dan sesudah diterapkan filter

Dari gambar 8 terlihat jelas bahwa adanya perbaikan data rotasi terhadap sumbu z sebelum dan sesudah diterapkan filter. Sebelum adanya filter terlihat data rotasi sangat fluktuatif hal tersebut akan menyebabkan terjadinya *jitter* pada saat dilakukan integrasi pada objek 3D, dengan adanya filter *moving average* hal tersebut dapat diminimalisir sehingga roatsi terhadap sumbu z menjadi lebih halus dan minim *jitter*.

Dari perbandingan yang terlihat pada grafik yang ditampilkan di gambar 5 sampai dengan gambar 8 dapat terlihat bahwa filter *moving average* cukup untuk mengurangi *jitter* yang terjadi saat adanya akuisisi data gerakan tangan menggunakan mediapipe. Akan tetapi pada beberapa bagian, filter ini juga tidak efektif dalam mengurangi terjadinya *jitter* atau bahkan menambah *jitter*. Terutama pada data rotasi gerakan tangan. Hal ini dapat terjadi karena data yang terdapat pada rotasi tangan sangat fluktuatif sehingga penggunaan filter *moving average* justru tidak banyak memperbaiki permasalahan yang terjadi. Jika dibandingkan hasil filter antara posisi tangan dengan rotasi tangan.

Pada penelitian ini juga dihitung rata-rata perbedaan, perbedaan maksimum, perbedaan minimum dan standar deviasi dari data sebelum dan sesudah penerapan filter. Data tersebut dapat dilihat pada tabel 3.

Tabel 3. Rata-rata Perbedaan, Perbedaan Maksimal, Perbedaan Minimal dan Standar Deviasi dari Data Sebelum dan Sesudah diterapkan Filter

	Mean Difference	max difference	min difference	standard deviation
Posisi X	0.02482	0.0664	0	0.02214
Posisi Y	0.008943959	0.056300169	0.050503057	0.028418887
Posisi Z	0.002114526	0.007624647	0.011140958	0.004944111
Rotasi X	4.031010017	448.37942	442.7681804	189.7255259
Rotasi Y	15.04123148	938.0240903	856.1368603	562.026488
Rotasi Z	30.84617637	535.8333138	590.370563	248.8440201

Pada tabel 3 terlihat bahwa penerapan filter *moving average* efektif untuk menghilangkan *jitter* pada gerakan tangan posisi x, y, dan z hal ini dibuktikan dengan rendahnya nilai rata-rata perubahan (*mean difference*), nilai perubahan terbesar (*max difference*), nilai perubahan terkecil (*min difference*) dan juga

standar defiasi. Sedangkan pada rotasi tangan nilai-nilai tersebut sangat besar, hal ini menandakan bahwa filter moving average tidak cukup efektif untuk menghilangkan *jitter* pada rotasi.

IV. KESIMPULAN

Filter *moving average* cukup efektif untuk menghilangkan *jitter* yang terjadi saat akuisisi gerakan tangan menggunakan mediapipe. Akan tetapi filter ini hanya efektif jika digunakan pada data posisi tangan saja, sedangkan pada rotasi tangan, filter ini tidak terlalu efektif. Hal ini dapat disebabkan oleh nilai rotasi tangan yang sangat fluktuatif. Perlu diadakan penelitian lanjutan menggunakan metode lainnya sehingga akan didapatkan hasil filter yang lebih baik dalam mengurangi *jitter* pada rotasi tangan.

REFERENSI

- [1] Z. Ren, J. Yuan, J. Meng, dan Z. Zhang, "Robust Part-Based Hand Gesture Recognition Using Kinect Sensor," *IEEE Trans. Multimed.*, vol. 15, no. 5, hal. 1110–1120, Agu 2013, doi: 10.1109/TMM.2013.2246148.
- [2] W. Yang, Z. Zhong, X. Zhang, L. Jin, C. Xiong, dan P. Wang, "Depth camera based real-time fingertip detection using multi-view projection," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8008 LNCS, no. PART 5, hal. 254–261, 2013, doi: 10.1007/978-3-642-39342-6_28.
- [3] F. Y. Rudi F, "Tacking Pergerakan Tangan Menggunakan Skeleton Tracking Kinect," *J. Artif. Intell. Softw. Eng.*, vol. 1, no. 2, hal. 67–70, 2021, doi: 10.30811/jaise.v1i2.2449.
- [4] F. Zhang *et al.*, "MediaPipe Hands: On-device Real-time Hand Tracking," Jun 2020, Diakses: Jun 13, 2025. [Daring]. Tersedia pada: <http://arxiv.org/abs/2006.10214>.
- [5] G. Casiez, N. Roussel, dan D. Vogel, "1€ filter: A simple speed-based low-pass filter for noisy input in interactive systems," *Conf. Hum. Factors Comput. Syst. - Proc.*, hal. 2527–2530, 2012, doi: 10.1145/2207676.2208639.
- [6] S. Hansun, "A new approach of moving average method in time series analysis," in *2013 Conference on New Media Studies (CoNMedia)*, Nov 2013, hal. 1–4, doi: 10.1109/CoNMedia.2013.6708545.