

Pengembangan Motion Capture Lima Jari Dengan Algoritma Complementary Filter

Development Of Five Finger Motion Capture With Complementary Filter Algorithm

Faizal Gifari¹, Anton Siswo Raharjo Ansori², Faisal Candrasyah Hasibuan³

^{1,2,3} Universitas Telkom, Bandung

faiszalgiefarree@students.telkomuniversity.ac.id¹, faicanhasfcb@telkomuniversity.ac.id²,

raharjo@telkomuniversity.ac.id³

Abstrak

Motion capture merupakan proses untuk memperoleh data yang menggambarkan gerakan manusia atau hewan. Gerak yang ditangkap merupakan gerakan-gerakan yang nyata untuk kemudian diaplikasikan ke dalam media digital. Terdapat dua jenis alat untuk MoCap, yaitu alat MoCap yang menggunakan penanda (marker) dan alat MoCap yang tanpa menggunakan penanda (markerless). Sensor IMU merupakan salah satu alat marker MoCap memiliki kelebihan dan kekurangan.

Pada penelitian ini, data gerakan direkam menggunakan sensor MPU-9250 dan Multiplexer untuk menghubungkan alamat I2C pada lima sensor MPU-9250. Data yang berupa nilai motion dikirimkan ke Firebase dan secara otomatis gerakan hasil konversi dibuat dalam bentuk skeleton (rig) dan berisi gerakan kelima jari, yang mana gerakan tersebut dapat digunakan sebagai bahan pembuatan animasi.

Berdasarkan dari hasil pengujian, simulasi perangkat keras MoCap lima jari berbasis sensor IMU, data sensor IMU menggunakan Complementary Filter dan tanpa menggunakan filter menghasilkan nilai error pitch, roll dan yaw. Nilai eror terkecil menggunakan Complementary Filter pada kelima jari diperoleh 0.06110% dan nilai error terkecil tanpa filter diperoleh sebesar 7.29331%.

Kata kunci: *Motion Capture*, I2C, MPU-9250, Firebase

Abstract

Motion capture is the process of obtaining data that describes the movement of humans or animals. The captured movement is a real movement which is then applied to digital media. There are two types of tools for MoCap, namely MoCap tools that use markers and MoCap tools that do not use markers (markerless). The IMU sensor is one of the MoCap marking tools that has advantages and disadvantages.

In this study, motion data was recorded using the MPU-9250 sensor and a multiplexer to connect the I2C address to the five MPU-9250 sensors. Data in the form of movement values is sent to Firebase and automatically converted movements are created in the form of a rig and contain five finger movements, the movements of which can be used as animation material.

Based on the test results, the five finger MoCap hardware simulation based on the IMU sensor, the IMU sensor data using a Complementary Filter and without using a filter produces pitch, roll and yaw error values. The smallest error value using the Complementary Filter on the five fingers is 0.06110% and the smallest error value without a filter is 7.29331%. Keywords: Vehicle Speed Detection, Frame Difference, Video Processing

Keywords: Motion Capture, I2C, MPU-9250, Firebase

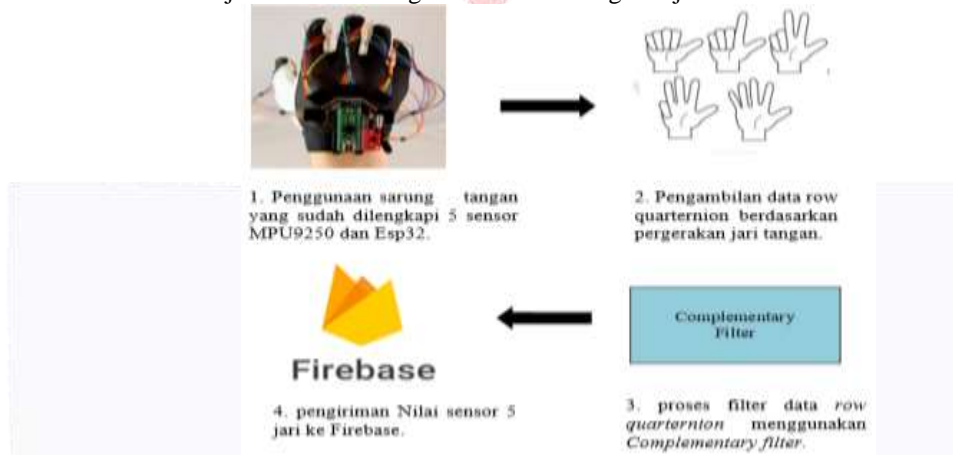
1. Pendahuluan

Motion capture atau sering disebut dengan MoCap merupakan bagian penting dari sebuah proses animasi baik dalam dunia film animasi 3D maupun dalam industri game [1]. Dalam implementasinya, terdapat dua jenis alat untuk MoCap, yaitu alat MoCap yang menggunakan penanda (marker) dan alat MoCap yang tanpa menggunakan penanda (markerless) [1]. Markerless MoCap merupakan teknik pengambilan gerakan manusia yang tidak menggunakan alat pada aktor dalam pakaian khususnya seperti modul kamera untuk teknik markerless MoCap [2]. Marker MoCap merupakan teknik pengambilan gerakan manusia, tetapi memerlukan alat atau pakaian khusus yang digunakan aktor. Teknik ini menggunakan sensor magnetis atau sensor flex untuk menentukan titik-titik sendi manusia.

Pada penelitian sebelumnya yang berjudul “Pengembangan Perangkat Keras Kontrol Busur Dan Anak Panah Berbasis Sensor IMU Dengan Metode *Complementary Filter* “, pada alat tersebut terdiri dari sebuah sensor IMU yang dipasang pada sarung tangan [3]. Penggunaan satu sensor sebagai alat marker MoCap dalam pengimplementasiannya hanya bisa menampilkan gerak bagian tangan saja sehingga pergerakan jari tangan tidak dapat ditampilkan di dalam animasi. Maka dari permasalahan tersebut dirancang motion capture lima jari dengan *algoritma complementary filter*. Dalam sistem ini dibutuhkan lima sensor MPU-9250 yang dipasangkan disetiap ujung jari tangan yang berfungsi sebagai penangkap gerak setiap jari dan penggunaan metode filtering pada data keluaran sensor cukup penting, dikarenakan terdapat noise atau gangguan yang dapat mengurangi keakurasian nilai keluaran dari sensor [4]. Maka, penggunaan metode filtering data dibutuhkan sebagai solusi dari permasalahan tersebut.

2. Perancangan Sistem

Sistem yang dibuat berdasarkan pergerakan kelima jari tangan pengguna, untuk pengerjaan sistem yang dibuat dibagi menjadi beberapa tahap perancangan yaitu, perancangan perangkat lunak maupun perancangan perangkat keras, pada perancangan perangkat keras, ESP32 akan dihubungkan dengan TCA9248A dan sensor MPU-9250 melalui pin I2C pada kedua *board* agar ESP32 dapat membaca nilai dari lima sensor MPU-9250 secara bersamaan, kemudian untuk perancangan perangkat lunak akan dibuat program filtrasi nilai sensor dengan metode *Complementary Filter* agar nilai yang dibaca dan dikirimkan tidak jauh berbeda dengan nilai referensi gerak jari.

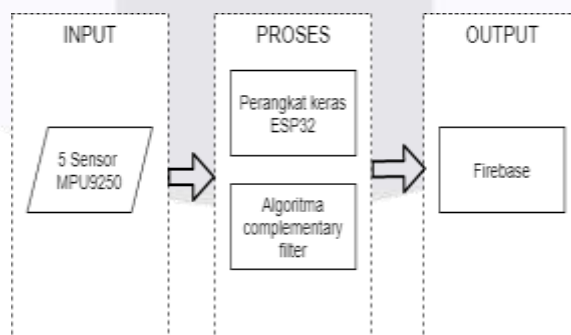


Gambar 2.1 Perancangan sistem

1. Sensor MPU-9250 akan tempatkan di setiap ujung jari tangan.
2. perangkat Esp2 akan menangkap dan mengolah data gerak jari yang dihasilkan dari sensor MPU-9250 pergerakan dari kedua tangan.
3. Data dari sensor MPU-9250 yaitu, *raw*, *pitch*, *yaw* akan di filter menggunakan metode *Complementary Filter*.
4. Data setelah di filter akan dikirim ke Firebase.

2.1 Analisis kebutuhan sistem

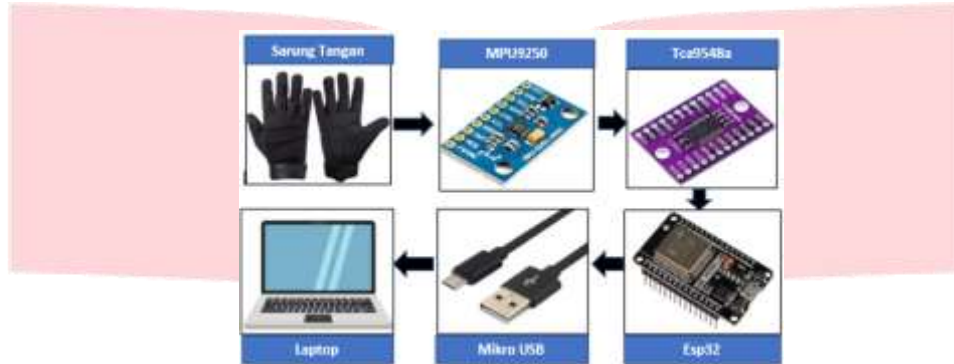
Pada analisis sistem prototipe sarung tangan ini membahas tentang hubungan rancang perangkat keras yang akan menghasilkan data, yang kemudian dikirim ke Firebase. Berikut merupakan gambar sistem secara keseluruhan:



Gambar 2.2 Analisis Sistem Prototipe Sarung Tangan

2.2 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras sangat dibutuhkan untuk mengetahui hubungan antar perangkat yang saling terhubung, agar tidak terjadi kesalahan pada jalur *input* dan *output* pada sistem. Berikut merupakan jalur *input* dan *output* sistem:

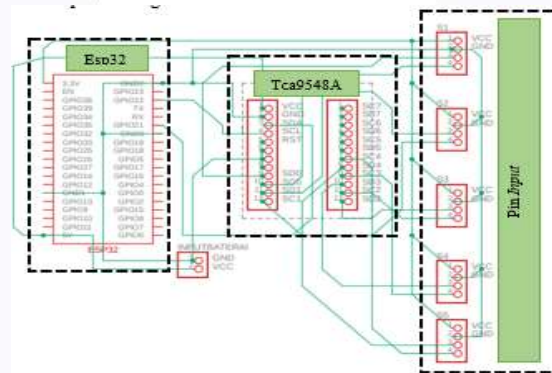


Gambar 2.3 Kebutuhan perangkat keras

Pemain akan menggunakan sarung tangan yang sudah dipasangkan ESP32 yang terhubung dengan TCA9548A dan lima sensor MPU-9250 di setiap jari, sensor MPU-9250 akan menangkap setiap pergerakan jari tangan tangan pemain, yang akan menghasilkan nilai *raw*, *pict*, *yaw*. Nilai yang dihasilkan dari pergerakan jari akan dijadikan sebagai kondisi pada virtualisasi jari tangan.

2.3 Skematik Mocap Lima Jari

Pada analisis sistem prototipe sarung tangan ini membahas tentang hubungan rancang perangkat keras yang akan menghasilkan data. Berikut merupakan gambar skematik sistem secara keseluruhan yang di rancang menggunakan aplikasi Eagle:



Gambar 2.4 skematik Mocap lima jari menggunakan aplikasi Eagle

Pada gambar 4 merupakan skematik keseluruhan sistem. Terdapat beberapa pin *input* yang berfungsi sebagai jalur penghubung lima sensor MPU9250 dan multiplexer Tca9548A. Pada bagian Esp32 terdapat dua pin tambahan yang berfungsi sebagai input baterai pada Esp32 yang terhubung langsung dengan pin *Vin* dan *ground* pada Esp32.

3. Hasil pengujian sistem

3.1 Hasil Pengujian Waktu Pengiriman Data Sensor IMU

Pengujian ini dilakukan dengan menghitung waktu dan jumlah data sensor pada saat dikirimkan ke Firebase.

Tabel 3.1 Hasil Pengujian Waktu Data Sensor IMU Menggunakan Complementary Filter

Parameter waktu pengiriman Data	Jumlah data terkirim dalam percobaan ke-			
	1	2	3	4
1 Menit	90 Data	105 Data	90 Data	105 Data
2 Menit	210 Data	225 Data	210 Data	210 Data
3 Menit	300 Data	315 Data	315 Data	315 Data

Tabel 3.2 Hasil Pengujian Waktu Data Sensor Tanpa filter

Parameter waktu pengiriman Data	Jumlah data terkirim dalam percobaan ke-			
	1	2	3	4
1 Menit	90 Data	105 Data	105 Data	105 Data
2 Menit	210 Data	195 Data	210 Data	210 Data
3 Menit	315 Data	330 Data	300 Data	315 Data

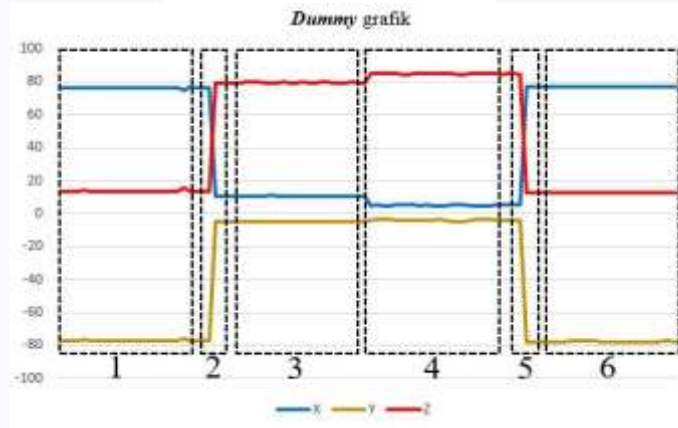
Pada Tabel 1 dan Tabel 2 di atas dapat dilihat rata-rata waktu pengiriman untuk setiap data sensor IMU DMP menggunakan *Complementary Filter* dan data sensor IMU DMP tanpa *filter* memiliki waktu sensor yang berbeda - beda yang dipengaruhi oleh kecepatan koneksi *Wi-Fi* yang digunakan.

3.2 Hasil Pengujian nilai sensor berdasarkan pergerakan jari

Pengujian ini dilakukan dengan menguji satu demi satu sensor MPU9250 dengan menggunakan *complementary filter* dan tanpa filter yang terpasang disetiap jari dengan empat kondisi, yaitu:

1. telapak tangan menghadap bawah (semua jari terbuka).
2. telapak tangan menghadap bawah (semua jari tertutup).
3. telapak tangan menghadap atas (semua jari terbuka).
4. telapak tangan menghadap atas (semua jari tertutup).

Pada pengujian ini dibuat *dummy* grafik yang bertujuan untuk memudahkan dalam memahami posisi sensor dan pergerakan kelima jari tangan.

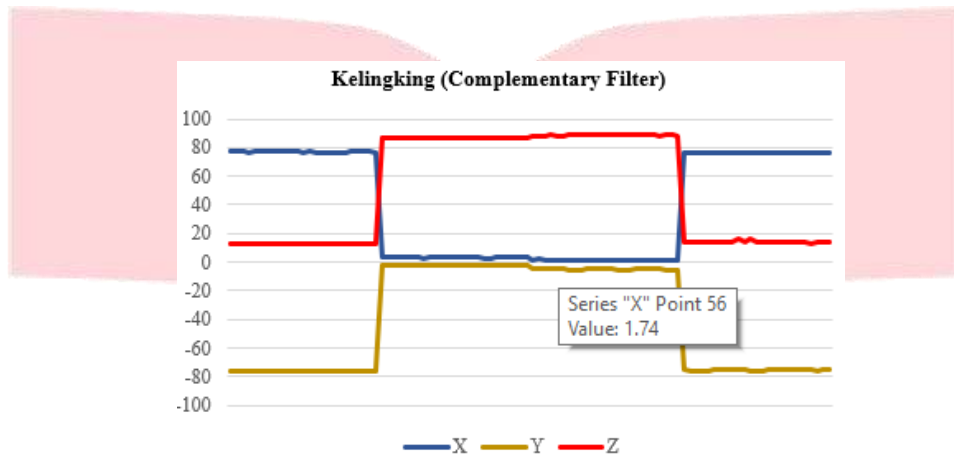


Gambar 3.1 Grafik dummy

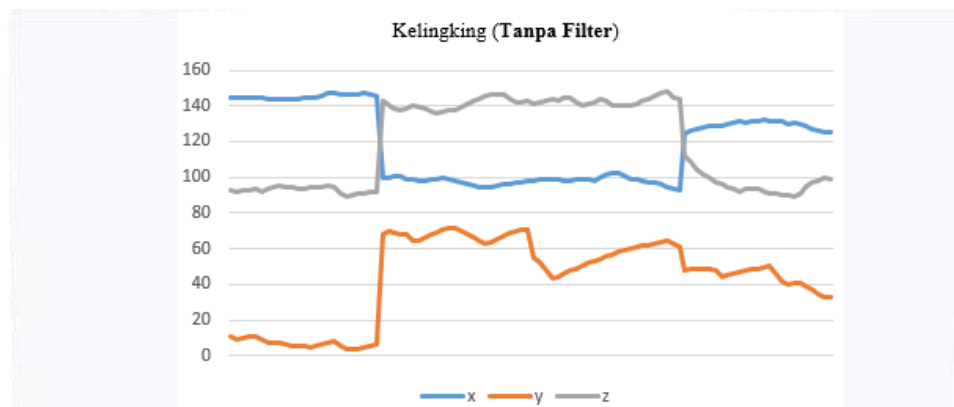
Gambar di atas merupakan grafik *dummy* pergerakan sensor pada jari dengan empat kondisi. Berikut penjelasannya:

1. Telapak tangan menghadap bawah (jari terbuka).
2. Perubahan posisi sensor (posisi jari akan menutup).
3. Telapak tangan menghadap bawah (jari tertutup).
4. Telapak tangan menghadap atas (jari terbuka).
5. Perubahan posisi sensor (jari akan menutup).
6. Telapak tangan menghadap atas (jari tertutup).

Perubahan posisi sensor ditandai dengan persilangan antara sumbu x dan sumbu Z. Berikut ini adalah plotting grafik pengujian nilai sensor disetiap jari tangan dengan *complementary filter* dan tanpa filter.

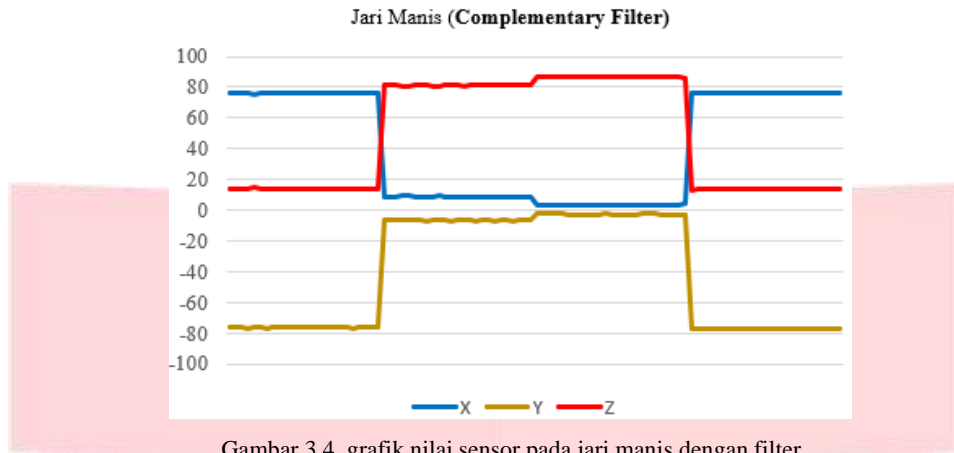


Gambar 3.2 grafik nilai sensor pada jari kelingking dengan filter

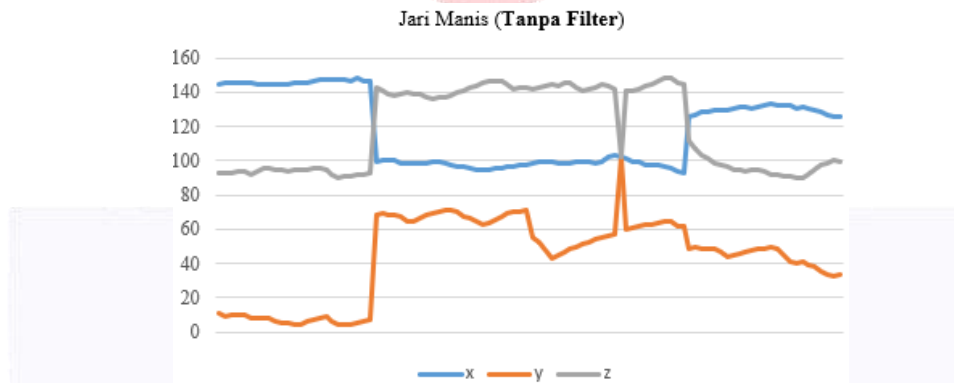


Gambar 3.3 grafik nilai sensor pada jari kelingking (tanpa) filter

Pada gambar 3.2 dan 3.3 adalah hasil *ploting* data dari sensor yang berada pada jari kelingking. Pada pengujian dengan filter, nilai x, y dan z didapatkan total eror sebesar 0,542 derajat dan pengujian tanpa filter total eror sebesar 7,419 derajat.

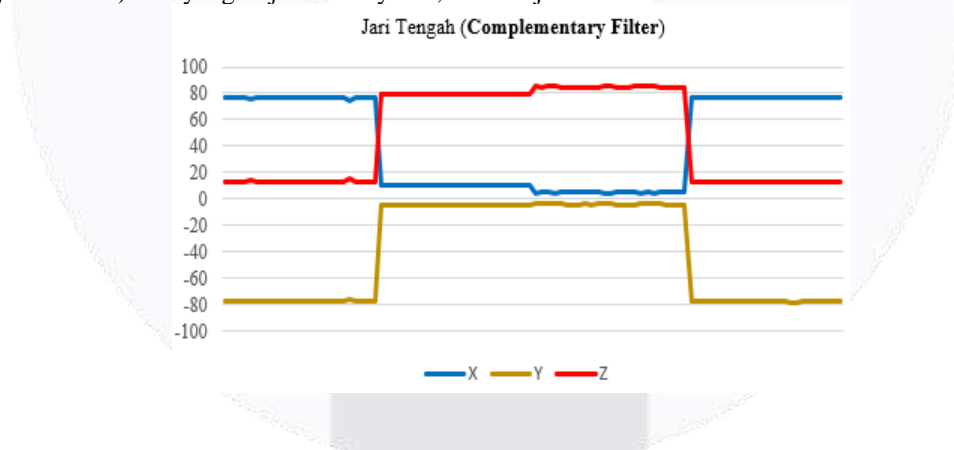


Gambar 3.4 grafik nilai sensor pada jari manis dengan filter

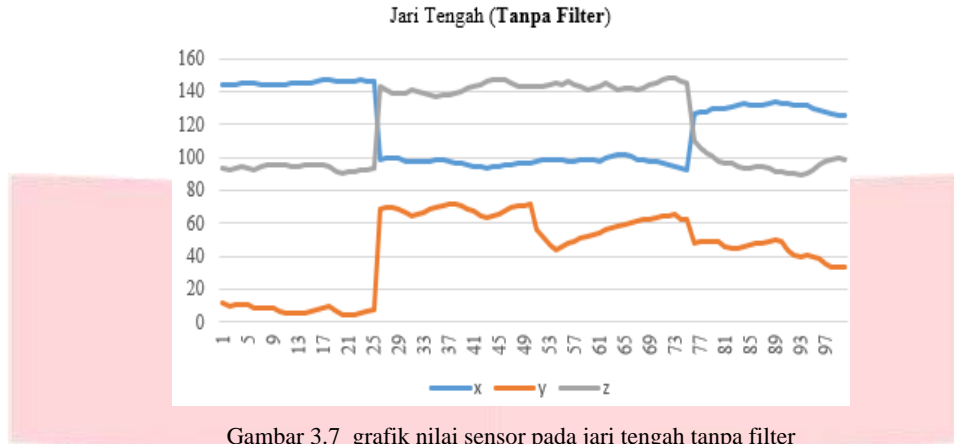


Gambar 3.5 grafik nilai sensor pada jari manis tanpa filter

Pada gambar 3.4 dan 3.5 adalah hasil *ploting* data dari sensor yang berada pada jari manis. Pada pengujian dengan filter, nilai x, y dan z didapatkan total eror sebesar 0,488 derajat dan pengujian tanpa filter total eror sebesar 9,555 derajat. Pada jari manis tanpa menggunakan filter total eror terbesar terjadi di sumbu y ketika posisi telapak tangan menghadap atas (semua jari terbuka) eror yang terjadi sebanyak 4,431 derajat.

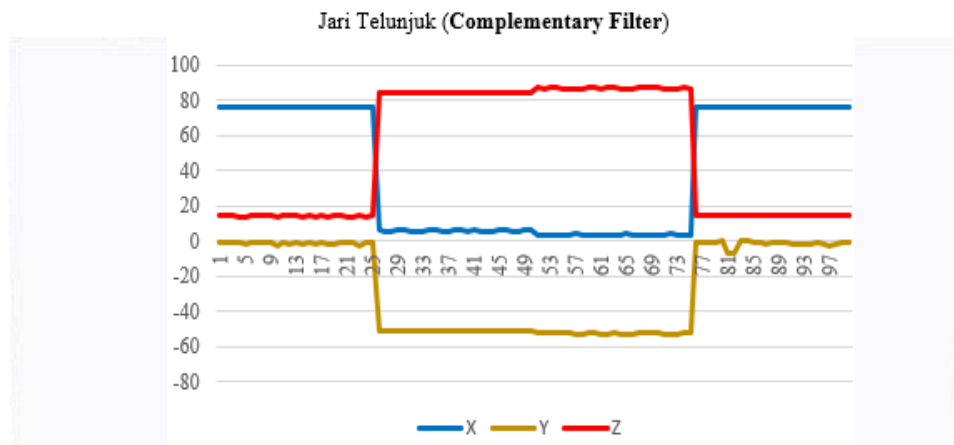


Gambar 3.6 grafik nilai sensor pada jari tengah dengan filter

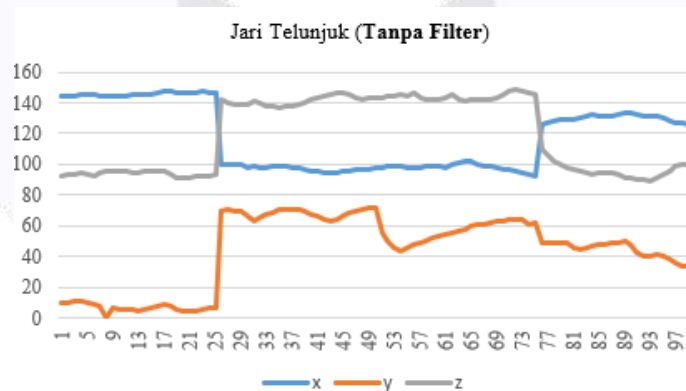


Gambar 3.7 grafik nilai sensor pada jari tengah tanpa filter

Pada gambar 3.6 dan 3.7 adalah hasil *ploting* data dari sensor yang berada pada jari tengah. Pada pengujian dengan filter, nilai x, y dan z didapatkan total eror sebesar 0,575derajat dan pengujian tanpa filter total eror sebesar 7,293derajat. Pada jari tengah tanpa menggunakan filter total eror terbesar terjadi di sumbu y ketika posisi telapak tangan menghadap atas (semua jari tertutup) eror yang terjadi sebanyak 2,621 derajat.

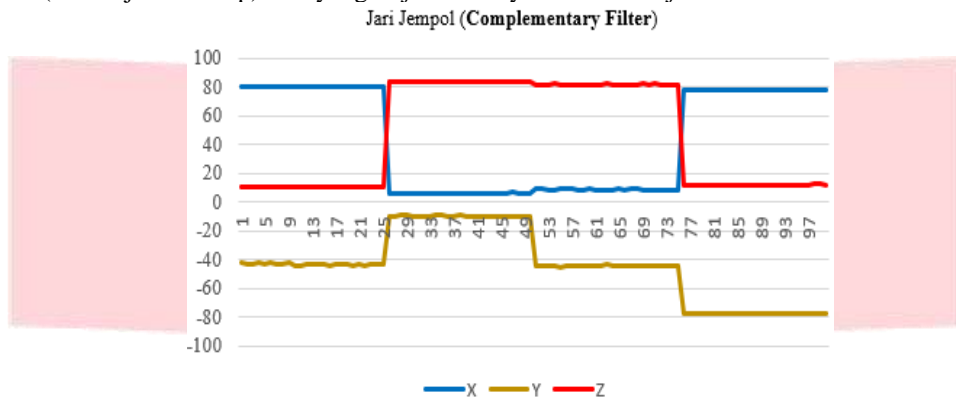


Gambar 3.8 grafik nilai sensor pada jari telunjuk dengan filter

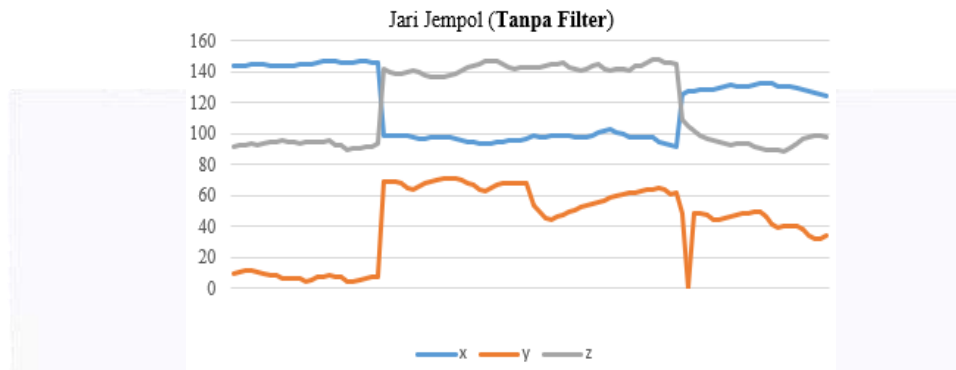


Gambar 3.9 grafik nilai sensor pada jari telunjuk tanpa filter

Pada gambar 3.8 dan 3.9 adalah hasil *ploting* data dari sensor yang berada pada jari telunjuk. Pada pengujian dengan filter, nilai x, y dan z didapatkan total eror sebesar 0,853 derajat dan pengujian tanpa filter total eror sebesar 7,279 derajat. Pada jari tengah tanpa menggunakan filter total eror terbesar terjadi di sumbu y ketika posisi telapak tangan menghadap atas (semua jari tertutup) eror yang terjadi sebanyak 2.631 derajat.



Gambar 3.10 grafik nilai sensor pada jari jempol dengan filter



Gambar 3.11 grafik nilai sensor pada jari jempol dengan filter

Pada gambar 3.10 dan 3.11 adalah hasil *ploting* data dari sensor yang berada pada jari jempol. Pada pengujian dengan filter, nilai x, y dan z didapatkan total eror sebesar 0,472 derajat dan pengujian tanpa filter total eror sebesar 7,347 derajat. Pada jari jempol dengan menggunakan filter peningkatan dan penurunan grafik pada sumbu y terjadi karena letak posisi sensor pada jari jempol berbeda (posisi sensor menghadap samping).

4. Kesimpulan dan Saran

4.1. Kesimpulan

Berdasarkan hasil uji coba sistem pada penelitian ini, didapatkan beberapa kesimpulan sebagai berikut:

1. Jumlah data yang terkirim ke Firebase menggunakan algoritma complementary filter dan tanpa filter sama, karena proses filtrasi tidak memperlambat pengiriman data.
2. Berdasarkan dari hasil pengujian, data sensor IMU menggunakan *Complementary Filter* total eror yang didapatkan dari pergerakan kelima jari sebesar 2,93 derajat, sedangkan tanpa menggunakan filter total nilai eror yang diperoleh dari pergerakan semua jari sebesar 38,971 derajat.

3. Pada pengujian parameter waktu pengiriman data sensor IMU yang dilakukan sebanyak 4 kali. Percobaan dengan menggunakan *Complementary Filter* dan tanpa filter waktu *delay* yang terbesar diperoleh 0,666 ms dan waktu *delay* terkecil diperoleh 0,545 ms.
4. Pada kinerja keseluruhan sistem prototipe sarung tangan untuk kesesuaian gerakan sensor IMU dengan player yang ada pada *Unity3D* memperoleh 100% kesuksesan dalam kesesuaian gerak.

4.2. Saran

1. Menggunakan sensor yang terbaru agar mendapatkan akurasi data yang lebih baik.
2. Menggunakan perangkat koneksi internet yang lebih cepat dan stabil agar mendapatkan waktu pengiriman data yang lebih kecil.

Referensi

- [1] Shingade, A., Ghotkar, A. Animation of 3D Human Model Using Markerless Motion Capture Applied To Sports. Jurnal ACM. 2014.
- [2] I Dewa, Ema Utami, Sukoco Optimalisasi Teknik Markerless Motion Capture Menggunakan Multisensor Pada Pembuatan Animasi 3D . Jurnal ACM. 2015.
- [3] Chandra, Development Of Hardware Control Bow And Arrow Based On IMU. 2020.
- [4] Furniss, Maureen, 2004. "Motion Capture", MIT Communications Forum.
- [5] Menache, Alberto, dan Morgan Kaufmann 2000. "Understanding motion capture for computer animation and video games".
- [6] M. Kitakawa and B. Windsor, "MoCap for Artists: workflow and techniques for motion capture," Massachusetts, USA.: Elsevier, 2008.
- [7] Usmart Charles Store, "New 2017 ESP32 Development Board WiFi+Bluetooth Ultra-Low Power Consumption Dual Cores ESP-32 ESP32S Hot Sale," Internet: <https://www.aliexpress.com/item/New-2017-ESP32-Development-Board-WiFiBluetooth-Ultra-Low-powerConsumption-Dual-Cores-ESP-32>, 2018. [Akses: 6 November 2020].
- [8] Ajie, S. 2016. Bekerja dengan I2C LCD dan Arduino. URL: <http://saptaji.com/2016/06/27/bekerja-dengan-I2C-lcd-dan-arduino/>. (Diakses pada 6 Oktober 2020) Gunawan, Ade Putra. (2012). Robot Keseimbangan Menggunakan Pengendali PID (Software).
- [9] A. Jiménez, F. Seco, J. Prieto, and J. Guevara, 2009. "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU," in Proc. IEEE Int. Symp. Intell. Signal Process., pp. 37–42.
- [10] H. J. Luinge and P. H. Veltink, 2004 "Inclination measurement of human movement using a 3-d accelerometer with autocalibration," vol. 12, pp. 112–121.