

PENGEMBANGAN MOUSE VIRTUAL MENGGUNAKAN SENSOR IMU MPU-9250 DAN ESP32 MENGGUNAKAN ALGORITMA DMP(DIGITAL MOTION PROCESSOR)

DEVELOPMENT OF VIRTUAL MOUSE BASED ON IMU SENSOR MPU-9250 AND ESP32 USING DMP ALGORITHM (DIGITAL MOTION PROCESSOR)

M Rifqi Nuril W K¹, Anton Siswo Raharjo Ansori², Ashri Dinimaharawati³

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

**wildanuril@student.telkomuniversity.ac.id¹, raharjo@telkomuniversity.ac.id²,
ashridini@telkomuniversity.ac.id³**

Abstrak

Pengembangan mouse virtual dapat dilakukan dengan membuat suatu desain yang melacak pergerakan tangan yang akan diubah menjadi pergerakan kursor, komponen yang diperlukan untuk pengembangan mouse virtual adalah sensor IMU yaitu sensor IMU MPU-9250 dengan menggunakan algoritma DMP(Digital Motion Processor) dan akan di filter menggunakan Complementary Filter untuk mengurangi noise dan akan diproses menggunakan mikrokontroler ESP32 dimana fungsi sensor untuk membaca pergerakan tangan kita yang akan diubah menjadi pergerakan kursor. Berdasarkan hasil pengujian yang di peroleh, pada saat simulasi prototipe mouse virtual data sensor MPU-9250 antara menggunakan complementary filter dan tanpa menggunakan complementary filter diperoleh nilai error pitch sebesar 0.40275% pada sudut 0°, 0.66575% pada sudut 30°, 0.4295% pada sudut 60°, dan 2.05425% pada sudut 90°, dan nilai error roll sebesar 0.1335% pada sudut 0°, 0.74% pada sudut 30°, 0.616% pada sudut 60°, dan 2.9515% pada sudut 90°, dan pada pengujian kesesuaian gerakan kursor pada prototipe mouse virtual memperoleh 100% kesuksesan dalam kesesuaian gerakan kursor.

Abstract

The development of a virtual mouse can be done by creating a design that tracks the movement of the hand which will be converted into cursor movement. Complementary Filter to reduce noise and will be processed using an ESP32 microcontroller where the sensor function is to read our hand movements which will be converted into cursor movements. Based on the test results obtained, when simulating the virtual mouse prototype MPU-9250 sensor data between using a complementary filter and without using a complementary filter, the error pitch value is 0.40275% at an angle of 0°, 0.66575% at an angle of 30°, 0.4295% at an angle of 60°, and 2.05425% at an angle of 90°, and roll error values of 0.1335% at an angle of 0°, 0.74% at an angle of 30°, 0.616% at an angle of 60°, and 2.9515% at an angle of 90°, and in testing the suitability of the cursor movement at virtual mouse prototype achieved 100% success in cursor movement compatibility.

Keywords: Mouse, ESP32, MPU-9250, Virtual, Complementary Filter.

1. Pendahuluan

Dengan pesatnya perkembangan teknologi hardware saat ini terutama pada mouse yang kita gunakan sehari-hari pada saat belajar maupun bekerja, di mana sekarang ini model serta fungsi mouse yang sangat beragam seperti mouse gaming yang memiliki fitur yang banyak seperti macro dan tidak lupa lampu RGB yang sangat memanjakan mata.

Tetapi mouse akan susah di gunakan pada tempat-tempat tertentu seperti saat kita sedang berada di bandara, stasiun maupun saat kita di kendaraan seperti mobil dan kereta api, mouse yang biasa kita pakai masih menggunakan alas sebagai tumpuan sehingga sensor dapat membaca pergerakan

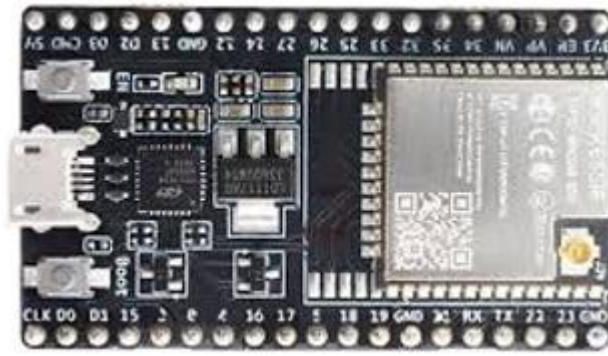
mouse sehingga dapat menggerakkan kursor, mouse akan susah digunakan secara normal karena kurangnya tempat pergerakan mouse.

Dari permasalahan diatas Mouse Virtual menggunakan sensor IMU MPU-9250 dengan menggunakan algoritma DMP(Digital Motion Processor) dan akan di filter menggunakan Complementary Filter untuk mengurangi noise, dengan begitu kita dapat menggunakan mouse dimanapun kita berada, hanya dengan menggerakkan telapak tangan maka kursor akan bekerja seperti kita menggunakan mouse yang biasanya kita pakai.

2. Dasar Teori

2.1. Mikrokontroler ESP32

Mikrokontroler ESP32 merupakan mikrokontroler yang di rancang oleh perusahaan Espressif Systems di Shanghai China. Mikrokontroler ESP32 dilengkapi dengan prosesor Xtensa dual core LX6, dengan tegangan 3.3 Volt, serta memiliki dukungan Wi-Fi dan BLE(Bluetooth Low Energy). Di lihat dari spesifikasi mikrokontroler ESP32 di atas dapat dijadikan pilihan untuk merancang mouse virtual karena memiliki dukungan WiFi yang tertanam pada mikrokontroler dan memiliki Bluetooth Low Energy juga untuk menghubungkan dari mikrokontroler ke komputer selain menggunakan kabel Micro USB[2].



Gambar 1 Gambar ESP32

2.2. MPU-9250

Sensor MPU-9250 merupakan sensor yang berfungsi untuk melacak pergerakan suatu objek, sensor MPU-9250 memiliki fitur DMP(Digital Motion Processing) yang bertujuan untuk mendapatkan nilai dari hasil sensing, pada sensor MPU-9250 dapat membaca nilai yaw, pitch, dan roll dalam bentuk raw atau mentah [3].



Gambar 2 Sensor MPU-9250

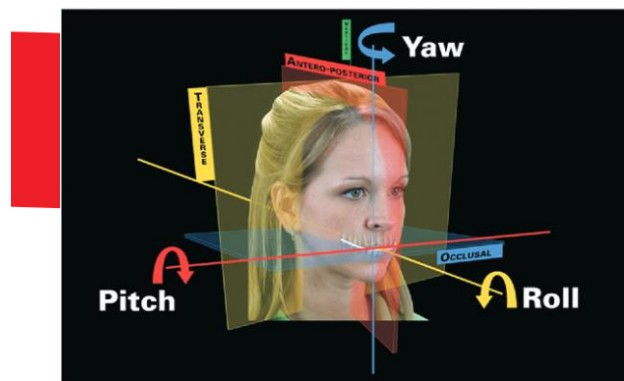
Pada gambar 2 diatas sensor MPU-9250 memiliki 9 DOF(Degree of Freedom) yaitu 3-axis Accelerometer (sensor percepatan), 3-axis Gyroscope (pengatur keseimbangan), dan 3-axis Magnetometer[4].

2.3. Motion Capture

Motion Capture atau yang lebih dikenal dengan MoCap merupakan perekaman atau penangkapan pergerakan suatu objek dimana pergerakan objek tersebut akan diubah kedalam bentuk model digital. Secara teknis pada *Motion Capture* adalah untuk mendapatkan sebuah data pada suatu titik tertentu dari pergerakan objek sehingga data yang didapatkan akan digunakan sebagai parameter tertentu seperti sudut, posisi, jarak, kecepatan, dan lain-lain. Sehingga data yang didapatkan akan dihitung dan dapat digunakan untuk mengontrol alat yang digunakan [5].

2.4. Nilai Pitch, Roll, dan Yaw

Nilai *Roll*, *Pitch*, dan *Yaw* di dapatkan dari arah pergerakan sensor contohnya nilai *roll* di dapat dari pergerakan sensor menghadap ke arah kanan/kiri, nilai *pitch* di dapatkan dari pergerakan sensor menghadap ke arah depan/belakang, dan nilai *yaw* di dapatkan dari perputaran arah sensor dari sisi ke sisi.

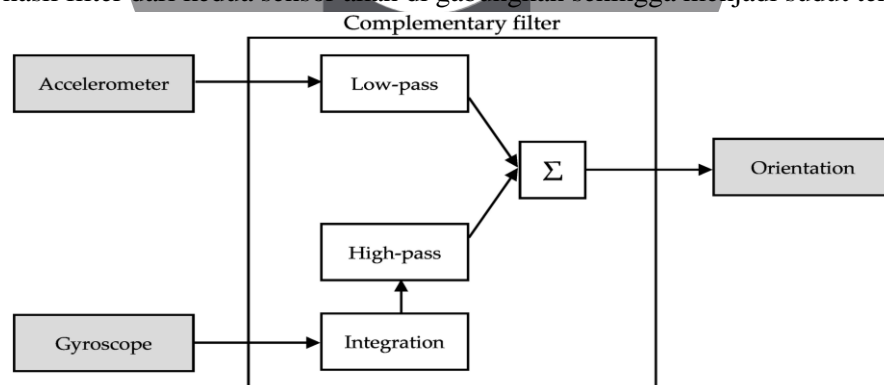


Gambar 3 Ilustrasi nilai roll, pitch, dan yaw[1].

Dari gambar 2.6 di atas merupakan ilustrasi wajah dapat bergerak dalam 3 bidang (depan/belakang, sisi ke sisi, dan atas dan ke bawah), dan juga dapat memutar sekitar 3 sumbu (horizontal, aksial, dan vertikal) yang di sebut sumbu roll, pitch, dan yaw[1].

2.5. Complementary Filter

Complementary Filter merupakan filter yang berfungsi untuk mengurangi noise pada output sensor Accelerometer dan sensor Gyroscope menggunakan Low Pass Filter dan High Pass Filter, kemudian hasil filter dari kedua sensor akan di gabungkan sehingga menjadi sudut tertentu [10].



Gambar 4 Diagram Complementary Filter

Berikut adalah rumus Complementary Filter:

$$x_angleC = a*(x_angleC + ang_gyro *dT) + (1-a) * (ang_accl) \quad (2.1)$$

dimana:

x_angleC = merupakan nilai Complementary

a = alpha

ang_gyro = output gyroscope

dT = delta time

ang_accl = output accelerometer

Sedangkan untuk mencari nilai alpha rumusnya sebagai berikut:

$$t = \frac{(a * dT)}{(1 - a)} \quad (2.2)$$

$$a = \frac{t}{(t + dT)} \quad (2.3)$$

dimana:

a = alpha

dT = waktu sampling

Dibawah ini merupakan rumus untuk mencari nilai rata-rata error pada setiap sudut :

$$\text{rata - rata error} = \frac{(\text{Jumlah banyak error})}{(\text{banyaknya error terjadi})} \quad (2.4)$$

Time constant(t) merupakan lamanya waktu update sinyal output hasil dari Complementary Filter [11][12].

3. Perancangan Sistem

3.1. Desain Sistem

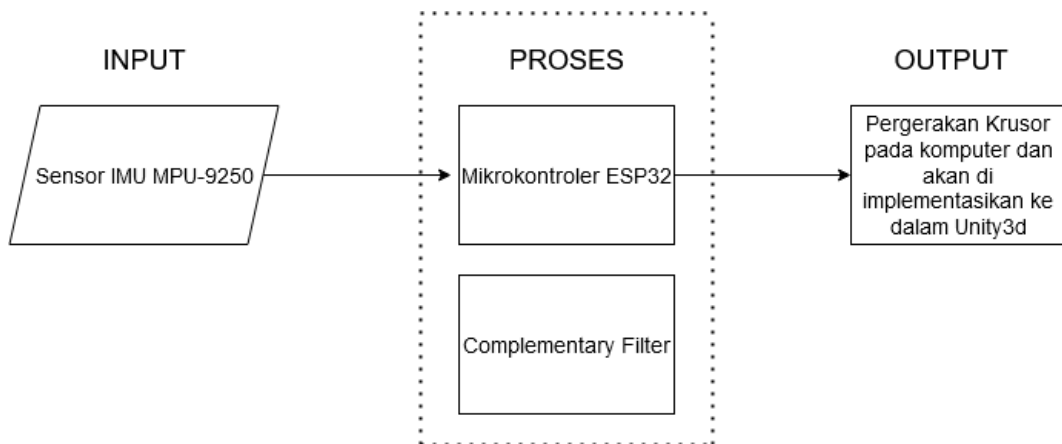
Desain sistem yang akan dirancang untuk membuat mouse virtual dengan cara mengambil data mentah hasil dari sensor IMU MPU-9250 yang nantinya akan menggerakkan kursor pada komputer. Desain yang akan digunakan menggunakan sensor IMU MPU-9250 sebagai alat yang akan menangkap gerak tangan kita yang nantinya akan dijadikan data mentah, lalu data akan di proses pada mikrokontroler ESP32 dan di filter menggunakan Complementary Filter. Hasil data yang sudah di filter menggunakan Complementary Filter akan akan menggerakkan kursor. Gambar Desain Sistem.



Gambar 5 Desain Sistem

3.2. Diagram Blok

Diagram blok berikut merupakan rancangan yang dibuat untuk mouse virtual dan membahas hubungan dari setiap perangkat. Berikut diagram blok secara keseluruhan:

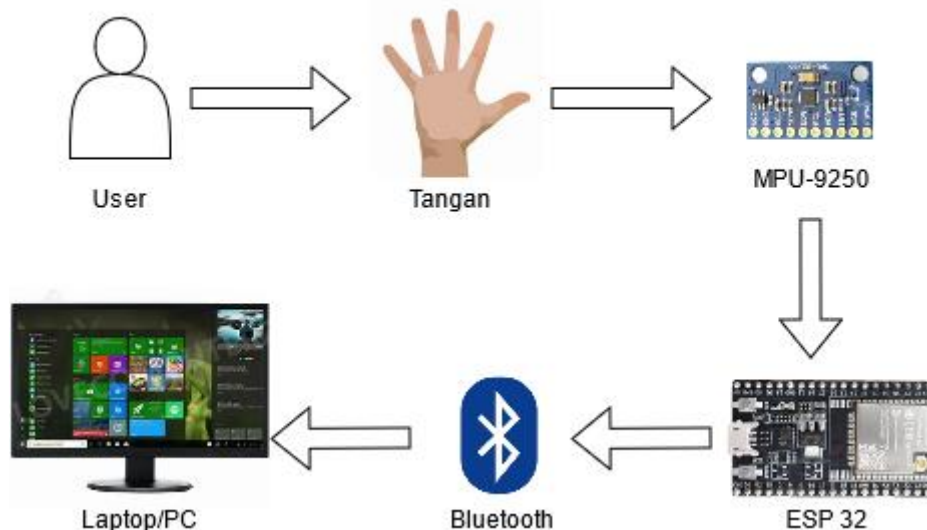


Gambar 6 Diagram Blok

Pada gambar 6 di atas merupakan cara kerja dari mouse virtual, pada sensor MPU-9250 akan mengambil data mentah lalu nilai kan di filter menggunakan Complementary Filter. Hasil data yang sudah di filter akan menghasilkan pergerakan kursor lalu hasil dari pergerakan kursor akan diimplementasikan kedalam monitor.

3.3. Desain Perangkat keras

Berikut merupakan bahasan mengenai perangkat keras yang akan digunakan pada penelitian ini sehingga antar perangkat saling terhubung.



Gambar 7 Desain Perangkat Keras

Pada Gambar 7 menjelaskan alur input dan output pada prototipe mouse virtual. User akan mengerjakan sensor menggunakan tangan yang sudah dipasang sensor MPU-9250 dan Mikrokontroler ESP32. Sensor MPU-9250 akan membaca pergerakan tangan pada sensor akan menghasilkan nilai pitch dan roll dan akan di filter menggunakan Complementary Filter.

Kemudian data yang telah di filter akan diproses menggunakan Mikrokontroler ESP32 dan akan di hubungkan ke Laptop/Pc menggunakan Bluetooth sehingga menjadi pergerakan kursor.

4. Hasil Pengujian dan Analisis

4.1. Hasil Pengujian Sensor MPU-9250

Hasil pengujian ini dilakukan pengujian sensor MPU-9250 dengan menggunakan Complementary Filter dan tanpa menggunakan Filter untuk mengetahui nilai sumbu X (roll), sumbu Y (pitch) dan sumbu Z (yaw) dengan sudut 0°, 30°, 60°, dan 90°. Berikut ini adalah tabel pengujian pada sensor MPU-9250 menggunakan Styrofoam yang sudah diukur menggunakan penggaris busur derajat. Berikut adalah tabel pengujian nilai error pada sumbu pitch dan roll, pada saat pengujian menggunakan Styrofoam yang sudah diukur menggunakan penggaris busur derajat.

Tabel 1 Tabel Pengujian Nilai Pitch.

Percobaan Ke-		Parameter Pengujian dalam Sudut (°)			
		0°	30°	60°	90°
1	Rata-rata Tanpa Menggunakan Filter	0.734	30.637	60.812	88.000
	Rata-rata Menggunakan Filter <i>Complementary Filter</i>	0.224	30.078	59.607	90.437
	Rata-rata Error	0.51	0.595	1.205	2.437
2	Rata-rata Tanpa Menggunakan Filter	0.745	30.248	60.505	87.767
	Rata-rata Menggunakan Filter <i>Complementary Filter</i>	0.325	29.532	60.464	89.444
	Rata-rata Error	0.42	0.716	0.041	2
3	Rata-rata Tanpa Menggunakan Filter	0.774	30.654	60.488	87.724
	Rata-rata Menggunakan Filter <i>Complementary Filter</i>	0.225	30.331	60.191	90.116
	Rata-rata Error	0.549	0.323	0.297	2.392
4	Rata-rata Tanpa Menggunakan Filter	0.755	30.803	60.471	87.716
	Rata-rata Menggunakan Filter <i>Complementary Filter</i>	0.623	29.774	60.296	89.104
	Rata-rata Error	0.132	1.029	0.175	1.388
Total Rata-rata Error Setiap Sudut		0.40275	0.66575	0.4295	2.05425

Pengujian yang dilakukan pada tabel 1 terhadap sumbu Y(*pitch*) menghasilkan perbedaan nilai yang tidak berbeda jauh, Dengan empat kali percobaan yang dilakukan mendapatkan nilai error dari sensor MPU-9250. Nilai error di dapatkan dari pengurangan hasil nilai dari pitch tanpa filter dengan nilai pitch yang telah di *filter* menggunakan *Complementary Filter*. Dari hasil pengujian, untuk menghitung rata-rata error pada setiap sudut menggunakan persamaan rumus (2.4).

$$\text{rata - rata error pada sudut } 0^\circ = \frac{0,51 + 0,42 + 0,549 + 0,132}{4} = 0.40275$$

$$\text{rata - rata error pada sudut } 30^\circ = \frac{0.595 + 0.716 + 0.323 + 1.029}{4} = 0.66575$$

$$\text{rata - rata error pada sudut } 60^\circ = \frac{1.205 + 0.041 + 0.297 + 0.175}{4} = 0.4295$$

$$\text{rata - rata error pada sudut } 90^\circ = \frac{2.437 + 2 + 2.392 + 1.388}{4} = 2.05425$$

Dari hasil penghitungan di atas mendapatkan nilai *error* rata-rata pada setiap sudut sensor MPU-9250 adalah 0.40275% pada sudut 0°, 0.66575% pada sudut 30°, 0.4295% pada sudut 60°, dan 2.05425% pada sudut 90°. Nilai *error* yang di dapat dari sensor MPU-9250 diakibatkan oleh posisi sensor dan nilai sudut yang dapat berubah-ubah, nilai *error* pada sensor MPU-9250 digunakan untuk melihat nilai sudut antara menggunakan *filter* atau tidak menggunakan filter. oleh karena itu nilai *error* yang dihasilkan oleh sensor MPU-9250 mendekati 0% sehingga sensor MPU-9250 bekerja dengan baik.

Tabel 2 Tabel Pengujian Nilai Roll.

Percobaan Ke-		Parameter Pengujian dalam Sudut (°)			
		0°	30°	60°	90°
1	Rata-rata Tanpa Menggunakan Filter	0.740	30.489	60.701	90.763
	Rata-rata Menggunakan Filter <i>Complementary Filter</i>	0.546	30.003	59.884	87.389
	Rata-rata Error	0.194	0.486	0.817	3.374
2	Rata-rata Tanpa Menggunakan Filter	0.428	30.740	60.813	90.833
	Rata-rata Menggunakan Filter <i>Complementary Filter</i>	0.530	29.916	59.930	88.099
	Rata-rata Error	0.102	0.824	0.883	2.734
3	Rata-rata Tanpa Menggunakan Filter	0.416	30.598	60.858	90.877
	Rata-rata Menggunakan Filter <i>Complementary Filter</i>	0.519	30.083	60.107	87.619
	Rata-rata Error	0.103	0.515	0.751	3.258
4	Rata-rata Tanpa Menggunakan Filter	0.411	30.606	60.072	90.593
	Rata-rata Menggunakan Filter <i>Complementary Filter</i>	0.546	29.471	60.059	88.153
	Rata-rata Error	0.135	1.135	0.013	2.44
Total Rata-rata Error Setiap Sudut		0.1335	0.74	0.616	2.9515

Pengujian yang dilakukan pada tabel 2 terhadap sumbu X(roll) menghasilkan perbedaan nilai yang tidak berbeda jauh, Dengan empat kali percobaan yang dilakukan mendapatkan nilai *error* dari sensor MPU-9250. Nilai *error* di dapatkan dari pengurangan hasil nilai dari pitch tanpa filter dengan nilai pitch yang telah di filter menggunakan *Complementary Filter*. Dari hasil pengujian, Untuk menghitung rata-rata *error* pada setiap sudut menggunakan persamaan rumus (2.4).

$$\text{rata - rata error pada sudut } 0^\circ = \frac{0.194 + 0.102 + 0.103 + 0.135}{4} = 0.1335$$

$$\text{rata - rata error pada sudut } 30^\circ = \frac{0.486 + 0.824 + 0.515 + 1.135}{4} = 0.74$$

$$\text{rata - rata error pada sudut } 60^\circ = \frac{0.817 + 0.883 + 0.751 + 0.013}{4} = 0.616$$

$$\text{rata - rata error pada sudut } 90^\circ = \frac{3.374 + 2.734 + 3.258 + 2.44}{4} = 2.9515$$

Dari hasil penghitungan di atas mendapatkan nilai *error* rata-rata pada setiap sudut sensor MPU-9250 adalah 0.1335% pada sudut 0°, 0.74% pada sudut 30°, 0.616% pada sudut 60°, dan 2.9515% pada sudut 90°. Nilai *error* yang di dapat dari sensor MPU-9250 diakibatkan oleh posisi sensor dan nilai sudut yang dapat berubah-ubah, nilai *error* pada sensor MPU-9250 digunakan

untuk melihat nilai sudut antara menggunakan filter atau tidak menggunakan filter, oleh karena itu nilai error yang dihasilkan oleh sensor MPU-9250 mendekati 0% sehingga sensor MPU-9250 bekerja dengan baik.

4.2. Hasil Pengujian Kesesuaian Gerakan Kursor

Berikut merupakan hasil pengujian kesesuaian gerakan kursor dimana pada saat arah sensor ke depan maka kursor akan ke arah atas, pada saat arah sensor ke belakang maka kursor akan ke arah bawah, pada saat arah sensor ke kanan maka kursor akan ke arah kanan, dan pada saat arah sensor ke arah kiri maka kursor akan ke arah kiri serta persentase keberhasilannya.

Tabel 3 Tabel Hasil Pengujian Gerakan Kursor.

Pengujian Ke-	Arah Sensor	Hasil Pengujian	Kesimpulan
1	Depan	Kursor Bergerak ke Arah Atas	Berhasil
	Belakang	Kursor Bergerak ke Arah Bawah	Berhasil
	Kanan	Kursor Bergerak ke Arah Kanan	Berhasil
	Kiri	Kursor Bergerak ke Arah Kiri	Berhasil
	Persentase Keberhasilan		100%
2	Depan	Kursor Bergerak ke Arah Atas	Berhasil
	Belakang	Kursor Bergerak ke Arah Bawah	Berhasil
	Kanan	Kursor Bergerak ke Arah Kanan	Berhasil
	Kiri	Kursor Bergerak ke Arah Kiri	Berhasil
	Persentase Keberhasilan		100%
3	Depan	Kursor Bergerak ke Arah Atas	Berhasil
	Belakang	Kursor Bergerak ke Arah Bawah	Berhasil
	Kanan	Kursor Bergerak ke Arah Kanan	Berhasil
	Kiri	Kursor Bergerak ke Arah Kiri	Berhasil
	Persentase Keberhasilan		100%
4	Depan	Kursor Bergerak ke Arah Atas	Berhasil
	Belakang	Kursor Bergerak ke Arah Bawah	Berhasil
	Kanan	Kursor Bergerak ke Arah Kanan	Berhasil
	Kiri	Kursor Bergerak ke Arah Kiri	Berhasil
	Persentase Keberhasilan		100%
Total Persentase Keberhasilan			100%

Pada Tabel 3 di atas dapat di lihat bahwa persentase pergerakan kursor menggunakan sensor MPU-9250 yaitu 100% sudah sesuai dan bekerja dengan baik.

4.3. Hasil Pengujian Kecepatan Kursor Pada Monitor

Berikut merupakan hasil pengujian menghitung waktu kursor bergerak pada kecepatan 1, 10, dan 20 DPI(Dots Per Inch) pada monitor dengan resolusi 1920 x 1080. Pengujian akan dilakukan menggunakan mouse biasa dan mouse virtual untuk melihat efisiensinya, pengujian akan dilakukan sebanyak 3 kali pada setiap mouse, pengujian dilakukan dengan cara menggerakkan mouse dengan kecepatan sedang.

Tabel 4 Tabel Pengujian Kursor Pada Kecepatan 1 DPI

No	Percobaan ke-	Parameter Hasil Pengujian			
		Dari Kiri ke Kanan	Dari Kanan ke Kiri	Dari Atas Ke Bawah	Dari Bawah ke atas
1	Menggunakan mouse biasa	10.05 detik	10.83 detik	8.21 detik	8.57 detik
2	Menggunakan mouse biasa	10.12 detik	11.06 detik	8.77 detik	8.77 detik
3	Menggunakan mouse biasa	10.50 detik	10.21 detik	8.87 detik	8.21 detik
Rata-rata		10.22 detik	10.7 detik	8.61 detik	8.51 detik
1	Menggunakan mouse virtual	46.44 detik	32.85 detik	58.22 detik	32.50 detik
2	Menggunakan mouse virtual	42.93 detik	30.92 detik	54.67 detik	38.61 detik
3	Menggunakan mouse virtual	48.14 detik	29.15 detik	56.24 detik	33.25 detik
Rata-rata		45.83 detik	30.97 detik	56.37 detik	34.78 detik

Pada Tabel 4 Nilai rata-rata yang di hasilkan pada pergerakan kursor mouse biasa menggunakan 1 DPI(Dots Per Inch) adalah 10.22 detik pada pergerakan kursor dari kiri ke kanan, 10.7 detik pada pergerakan kursor dari kanan ke kiri, 8.61 detik pada pergerakan kursor dari atas ke bawah, dan 8.51 detik pada pergerakan kursor dari bawah ke atas.

Nilai rata-rata yang di hasilkan pada pergerakan kursor mouse virtual menggunakan 1 DPI(Dots Per Inch) adalah 45.83 detik pada pergerakan kursor dari kiri ke kanan, 30.97 detik pada pergerakan kursor dari kanan ke kiri, 56.37 detik pada pergerakan kursor dari atas ke bawah, dan 34.78 detik pada pergerakan kursor dari bawah ke atas.

Tabel 5 Tabel Pengujian Kursor Pada Kecepatan 10 DPI

No	Percobaan ke-	Parameter Hasil Pengujian			
		Dari Kiri ke Kanan	Dari Kanan ke Kiri	Dari Atas Ke Bawah	Dari Bawah ke atas
1	Menggunakan mouse biasa	1.69 detik	1.65 detik	1.15 detik	1.22 detik
2	Menggunakan mouse biasa	1.59 detik	1.62 detik	1.12 detik	1.14 detik
3	Menggunakan mouse biasa	1.46 detik	1.59 detik	1.22 detik	1.22 detik
Rata-rata		1.58 detik	1.62 detik	1.16 detik	1.19 detik
1	Menggunakan mouse virtual	3.85 detik	1.59 detik	1.91 detik	1.79 detik
2	Menggunakan mouse virtual	3.69 detik	1.48 detik	1.75 detik	1.75 detik
3	Menggunakan mouse virtual	3.63 detik	1.74 detik	1.99 detik	1.95 detik
Rata-rata		3.72 detik	1.60 detik	1.88 detik	1.83 detik

Pada Tabel 5 Nilai rata-rata yang di hasilkan pada pergerakan kursor mouse biasa menggunakan 10 DPI(Dots Per Inch) adalah 1.58 detik pada pergerakan kursor dari kiri ke kanan, 1.62 detik pada pergerakan kursor dari kanan ke kiri, 1.16 detik pada pergerakan kursor dari atas ke bawah, dan 1.19 detik pada pergerakan kursor dari bawah ke atas.

Nilai rata-rata yang di hasilkan pada pergerakan kursor mouse virtual menggunakan 10 DPI(Dots Per Inch) adalah 3.72 detik pada pergerakan kursor dari kiri ke kanan, 1.60 detik pada pergerakan kursor dari kanan ke kiri, 1.88 detik pada pergerakan kursor dari atas ke bawah, dan 1.83 detik pada pergerakan kursor dari bawah ke atas.

Tabel 6 Tabel Pengujian Kursor Pada Kecepatan 20 DPI

No	Percobaan ke-	Parameter Hasil Pengujian			
		Dari Kiri ke Kanan	Dari Kanan ke Kiri	Dari Atas Ke Bawah	Dari Bawah ke atas
1	Menggunakan mouse biasa	0.65 detik	0.53 detik	0.47 detik	0.40 detik
2	Menggunakan mouse biasa	0.53 detik	0.53 detik	0.37 detik	0.38 detik
3	Menggunakan mouse biasa	0.57 detik	0.62 detik	0.44 detik	0.38 detik
Rata-rata		0.58 detik	0.56 detik	0.42 detik	0.38 detik
1	Menggunakan mouse virtual	1.18 detik	0.71 detik	0.74 detik	0.90 detik
2	Menggunakan mouse virtual	0.87 detik	0.57 detik	0.78 detik	0.68 detik
3	Menggunakan mouse virtual	0.62 detik	0.59 detik	0.72 detik	0.87 detik
Rata-rata		0.89 detik	0.62 detik	0.74 detik	0.81 detik

Pada Tabel 6 Nilai rata-rata yang di hasilkan pada pergerakan kursor mouse biasa menggunakan 20 DPI(Dots Per Inch) adalah 0.58 detik pada pergerakan kursor dari kiri ke kanan, 0.56 detik pada pergerakan kursor dari kanan ke kiri, 0.42 detik pada pergerakan kursor dari atas ke bawah, dan 0.38 detik pada pergerakan kursor dari bawah ke atas.

Nilai rata-rata yang di hasilkan pada pergerakan kursor mouse virtual menggunakan 20 DPI(Dots Per Inch) adalah 0.89 detik pada pergerakan kursor dari kiri ke kanan, 0.62 detik pada pergerakan kursor dari kanan ke kiri, 0.74 detik pada pergerakan kursor dari atas ke bawah, dan 0.81 detik pada pergerakan kursor dari bawah ke atas.

Pada tabel 4, 5, dan 6 perbedaan pada pergerakan kursor menggunakan mouse biasa dan mouse virtual tidak terlalu signifikan keduanya sama-sama sensitif dan efisien saat digunakan.

5. Kesimpulan dan Saran

5.1. Kesimpulan

1. Hasil pengujian yang diperoleh pada saat simulasi prototipe mouse virtual data sensor MPU9250 menggunakan complementary filter dan tanpa menggunakan complementary filter diperoleh nilai error pitch sebesar 0.40275 derajat pada sudut 0°, 0.66575 derajat pada sudut 30°, 0.4295 derajat pada sudut 60°, dan 2.05425 derajat pada sudut 90°, dan nilai error roll sebesar 0.1335 derajat pada sudut 0°, 0.74 derajat pada sudut 30°, 0.616 derajat pada sudut 60°, dan 2.9515 derajat pada sudut 90°.
2. Hasil pengujian kesesuaian gerakan kursor pada prototipe mouse virtual memperoleh 100% kesuksesan dalam gerakan kursor.
3. Hasil pengujian kecepatan kursor pada monitor dengan resolusi 1920 x 1080 diperoleh nilai rata-rata yang di hasilkan pada pergerakan kursor mouse biasa menggunakan 1 DPI(Dots Per Inch) adalah 10.22 detik pada pergerakan kursor dari kiri ke kanan, 10.7 detik pada pergerakan kursor dari kanan ke kiri, 8.61 detik pada pergerakan kursor dari atas ke bawah, dan 8.51 detik pada pergerakan kursor dari bawah ke atas. Nilai rata-rata yang di hasilkan pada pergerakan kursor mouse virtual menggunakan 1 DPI(Dots Per Inch) adalah 45.83 detik pada pergerakan kursor dari kiri ke kanan, 30.97 detik pada pergerakan kursor dari kanan ke kiri, 56.37 detik pada pergerakan kursor dari atas ke bawah, dan 34.78 detik pada pergerakan kursor dari bawah ke atas. Nilai rata-rata yang di hasilkan pada pergerakan kursor mouse biasa menggunakan 10 DPI(Dots Per Inch) adalah 1.58 detik pada pergerakan kursor dari kiri ke kanan, 1.62 detik pada pergerakan kursor dari kanan ke kiri, 1.16 detik pada pergerakan kursor dari atas ke bawah, dan 1.19 detik pada pergerakan kursor dari bawah ke atas. Nilai rata-rata yang di hasilkan pada pergerakan kursor mouse virtual menggunakan 10 DPI(Dots Per Inch) adalah 3.72 detik pada pergerakan kursor dari kiri ke kanan, 1.60 detik pada pergerakan kursor dari kanan ke kiri, 1.88 detik pada pergerakan kursor dari atas ke bawah, dan 1.83 detik pada pergerakan kursor dari bawah ke atas. Nilai rata-rata yang di hasilkan pada pergerakan kursor mouse biasa menggunakan 20 DPI(Dots Per Inch) adalah 0.58 detik pada pergerakan kursor dari kiri ke kanan, 0.56 detik pada pergerakan kursor dari kanan ke kiri, 0.42 detik pada pergerakan kursor dari atas ke bawah, dan 0.38 detik pada pergerakan kursor dari bawah ke atas. Nilai rata-rata yang di hasilkan pada pergerakan kursor mouse virtual menggunakan 20 DPI(Dots Per Inch) adalah 0.89 detik pada pergerakan kursor dari kiri ke kanan, 0.62 detik pada pergerakan kursor dari kanan ke kiri, 0.74 detik pada pergerakan kursor dari atas ke bawah, dan 0.81 detik pada pergerakan kursor dari bawah ke atas.

5.2. Saran

Berdasarkan hasil perancangan dan pengujian sistem yang telah dilakukan pada tugas akhir ini, maka berikut adalah saran-saran yang diusulkan untuk penelitian lebih lanjut, yaitu:

1. Menambahkan button untuk prototipe mouse virtual mouse agar lebih efisien.
2. Menggunakan sensor terbaru agar mendapatkan akurasi data yang lebih baik.
3. Menggunakan mikrokontroler yang terbaru agar mendapatkan fitur yang lebih banyak.

Referensi:

- [1] J. L. Ackerman, W. R. Proffit, D. M. Sarver, M. B. Ackerman, and M. R. Kean, "Pitch, roll, and yaw: Describing the spatial orientation of dentofacial traits," *Am. J. Orthod. Dentofac. Orthop.*, vol. 131, no. 3, pp. 305–310, 2007, doi: 10.1016/j.ajodo.2006.05.032.
- [2] Muliadi, A. Imran, and M. Rasul, "Pengembangan tempat sampah pintar menggunakan esp32," *MEDIA Elektr.*, vol. 17, no. 2, pp. 1907–1728, 2020.
- [3] B. Darmawan, D. Syauqy, M. Hannats, and H. Ichsan, "Rancang Bangun View controller Menggunakan Sensor Akselerometer Pada Game Bus Simulator Sebagai Sarana User Experience (UX) Berbasis Sistem Embedded," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 2, pp. 1883–1891, 2019.
- [4] B. Firman, "Implementasi Sensor IMU MPU6050 Berbasis Serial I2C pada Self-Balancing Robot," *J. Teknol. Technoscintia*, vol. 9, no. 1, pp. 18–24, 2016.
- [5] S. M Idham Rusdi, M. Suyatno, "ANIMASI 3D EKSPRESI WAJAH MENGGUNAKAN TEKNIK MARKERLESS MOTION CAPTURE," *Ilm. d'Computare*, vol. 8, pp. 30–41, 2018.
- [6] M. Riyadi, Wahyudi, and I. Setiawan, "Pendeteksi Posisi Menggunakan Sensor Accelerometer MMA7260Q Berbasis Mikrokontroler ATMEGA 32," *Transmisi*, vol. 12, no. 2, pp. 76–81, 2010, doi: 10.12777/transmisi.12.2.76-81.
- [7] F. Barrera, R. Sampaio, D. Munoz, and C. H. Llanos, "A Study of Attitude and Heading Determination through an EKF-based Sensor Fusion for Inertial Measurement Units (IMUs)," *24th ABCM Int. Congr. Mech. Eng.*, 2017, doi: 10.26678/abcm.cobem2017.cob17-1859.
- [8] W. A. Kusuma, Z. Sari, and A. T. Sari, "Sensor Fusion Accelerometer dan Gyroscope untuk Pengukuran Perubahan Kinematik Pergelangan Kaki," *Kinetik*, vol. 1, no. 1, pp. 17–22, 2016, doi: 10.22219/kinetik.v1i1.8.
- [9] E. S. Noor, M. Rif, P. Siwindarto, and M. E. Sc, "Implementasi Sensor Magnetometer dan Akselerometer," *J. Mhs. TEUB*, vol. 1, no. 2, pp. 1–6, 2013.
- [10] R. Bimarta, A. Eko Putra, and A. Dharmawan, "Balancing Robot Menggunakan Metode Kendali Proporsional Integral Derivatif," *IJEIS*, vol. 5, no. 1, pp. 89–98, 2015.
- [11] H. Z. Fahmi, R. Maulana, and W. Kurniawan, "Implementasi Complementary Filter Menggunakan Sensor Accelerometer dan Gyroscope pada Keseimbangan Gerak Robot Humanoid," *Pengemb. Teknol. Inf. dan Ilmu Komput*, vol. 1, no. 11, pp. 1376–1384, 2017.
- [12] L. Yang, J. Ren, K. Song, X. Ma, and X. Yang, "Complementary Filter Design Based Error Allocation for Attitude Estimation with Low-Cost Sensor," *Adv. Eng. Res.*, vol. 140, no. Ecae 2017, pp. 206–211, 2018.