

LAMPIRAN V

PENGEMBANGAN PERANGKAT KERAS KONTROL BUSUR DAN ANAK PANAH BERBASIS SENSOR IMU MENGGUNAKAN METODE KALMAN FILTER

DEVELOPMENT OF HARDWARE CONTROL BOW AND ARROW BASED ON IMU SENSOR USING KALMAN FILTER METHOD

Farel Mohammad Fauzi^{#1}, Fery Chandra^{#2}

^{1,2}Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

¹farelft@student.telkomuniversity.ac.id, ²ferychandra@student.telkomuniversity.ac.id

Abstrak

Panahan adalah permainan yang menggunakan busur sebagai media untuk menembakan anak panah menuju target. Panahan ini sangat digemari dikarenakan dapat membantu dalam meningkatkan keseimbangan tubuh dan meningkatkan koordinasi tangan dan mata. Dengan teknologi pengolahan citra yang sudah berkembang, sekarang permainan panahan tidak membutuhkan tempat yang besar untuk memainkannya. Dengan menggunakan teknologi *Virtual Reality* yang disimulasikan menggunakan komputer (*computer-simulated environment*), pengguna dapat memainkan panahan dengan menciptakan ilusi tiga dimensi yang realistis dengan menggunakan teknik – teknik grafis komputer. Tugas akhir ini bertujuan untuk membuat simulasi sistem untuk memainkan permainan panahan dengan membuat prototipe sarung tangan untuk mengontrol busur dan anak panah menggunakan sensor MPU9250 dan perangkat NodeMCU dengan menggunakan metode *Kalman Filter*.

Kata kunci : Panahan, Busur, Anak Panah, MPU9250, NodeMCU, *Kalman Filter*

Abstract

Archery is a game that uses a bow as a medium to shoot arrows at targets. This archery is very popular because it can help in increasing body balance and improving hand and eye coordination. With image processing technology that has developed, now the archery game does not require a large place to play. By using Virtual Reality technology that is simulated using a computer (*computer-simulated environment*), users can play archery by creating realistic three-dimensional illusions using computer graphics techniques. This final project aims to create a system simulation to play an archery game by making a prototype of a glove to control the bow and arrow using the MPU9250 sensor and the NodeMCU device using the Kalman Filter method.

Keywords: Archery, Bows, Arrows, MPU9250, NodeMCU, Kalman Filter

1. Pendahuluan

Panahan atau dalam Bahasa Inggris dinamakan *Archery* merupakan permainan yang menggunakan busur panah dan anak panah sebagai pengaplikasiannya, dengan menembakan anak panah melalui busur panah untuk mengenai sasaran dengan jarak tertentu [1]. Permainan panahan ini sangat digemari dikarenakan dapat membantu dalam meningkatkan keseimbangan tubuh dan meningkatkan koordinasi tangan dan mata.

Virtual Reality (VR) atau realitas maya adalah teknologi pengolahan citra yang disimulasikan menggunakan komputer (*computer-simulated environment*), sehingga pengguna dapat berinteraksi dengan suatu lingkungan yang menggabungkan dua konsep yaitu virtual dan realitas [2]. Teknologi virtual reality ini mampu menganalisis dan menciptakan berbagai realitas yaitu dengan menciptakan ilusi tiga dimensi yang realistis dengan menggunakan teknik – teknik grafis komputer.

Terdapat permasalahan yang membuat permainan panahan sulit untuk dimainkan, dikarenakan permainan panahan membutuhkan tempat yang besar untuk dimainkan, karena pada permainan panahan memiliki beberapa kategori dalam jarak tembak ke target yaitu 18 meter sampai dengan 90 meter dengan lebar 4,5 meter sampai 5 meter. Sehingga sangat sulit untuk menemukan tempat dengan panjang dan lebar lapangan yang sesuai di daerah perkotaan [3].

Berdasarkan permasalahan di atas dibutuhkan solusi untuk menyelesaikan permasalahan tersebut yaitu dengan membuat perangkat keras kontrol busur dan anak panah berbasis sensor IMU menggunakan metode algoritma *Kalman filter*. Kontrol busur dan anak panah berbasis sensor IMU akan diimplementasikan kedalam *Unity* menggunakan *VR* sehingga tidak membutuhkan tempat yang besar untuk melakukan kegiatan panahan.

2. Perancangan Sistem

2.1 Gambaran Umum Sistem

Sistem yang dibuat berdasarkan pergerakan kedua tangan pengguna yang akan menggerakkan karakter dan melepaskan anak panah didalam game panahan pada aplikasi *Unity3d* menggunakan metode *Kalman Filter*. Simulasi sistem ini dibuat dengan tujuan untuk memainkan game panahan menggunakan prototipe sarungan tangan menggunakan sensor MPU9250.

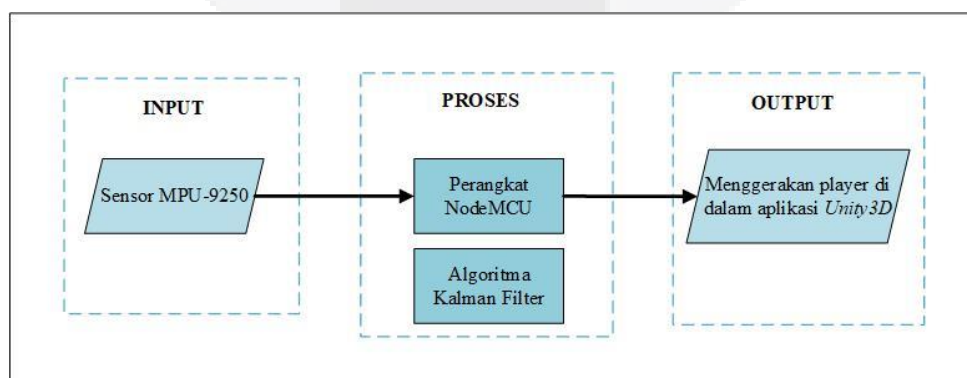


Gambar 1 Gambaran Umum Sistem

Pada Gambar 1 menggambarkan tentang gambaran umum dari prototipe sarung tangan yaitu 1. Penggunaan prototipe sarung tangan pada kedua tangan. 2. Sensor MPU9250 dan Perangkat NodeMCU akan menangkap pergerakan dari kedua tangan. 3. Data dari sensor MPU9250 yaitu *raw quaternion* akan di filter menggunakan metode *Kalman filter*. 4. Data setelah di filter akan dikirimkan kedalam *Unity3D* untuk menggerakkan player dan melepaskan anak panah pada game panahan.

2.2 Analisis Kebutuhan Sistem

Pada analisis sistem protatipe sarung tangan ini membahas hubungan rancang perangkat keras yang akan menghasilkan data untuk dapat di akses oleh aplikasi *Unity3D*. Berikut merupakan gambar sistem secara keseluruhan :



Gambar 2 Analisis Sistem Prototipe Sarung Tangan

Pada Gambar 2 menjelaskan bagaimana prototipe sarung tangan dapat menggerakkan player dan anak panah di dalam *Unity3D*. Sensor MPU9250 akan menangkap pergerakan dari kedua tangan, Kemudian nilai yang dihasilkan akan di *filter* menggunakan metode *Kalman Filter*. Data yang telah di *filter* akan di akses oleh *Unity3D* melalui perangkat NodeMCU.

2.2.1 Skematik Sensor MPU9250

Fungsi dari sensor MPU9250 adalah untuk menghasilkan data dari pergerakan kedua tangan yang akan diakses oleh NodeMCU. Skematik sensor MPU9250 sebagai berikut :

Tabel 1 Skematik Sensor MPU9250

No	Nama Pin	Connect to
1	VCC	VCC NodeMCU
2	SCL	D1 NodeMCU
3	SDA	D2 NodeMCU
4	GND	GND NodeMCU

Pada Tabel 1 menunjukan skematik dari sensor MPU9250. Sensor MPU9250 akan terhubung dengan NodeMCU dengan tagangan 3v sebagai *power supply*.

2.2.2 Skematik NodeMCU

Fungsi dari perangkat NodeMCU adalah untuk mengirimkan data yang didapatkan dari sensor MPU9250 untuk dapat di akses oleh *Unity3D*. Skematik dari perangkat NodeMCU sebagai berikut :

Tabel 2 Skematik NodeMCU

No	Nama Pin	Connect to
1	VCC	VCC MPU9250
2	D1	SCL MPU9250
3	D2	SDA MPU9250
4	GND	GND MPU9250

Pada Tabel 2 menunjukan skematik perangkat NodeMCU. Kedua pin yang digunakan terhubung dengan sensor MPU9250, perangkat NodeMCU akan mengirimkan data untuk menggerakkan karakter pada game panahan didalam *Unity3d*.

3. Hasil Pengujian Sistem

3.1 Hasil Pengujian Sensor Sudut MPU9250

Pengujian ini dilakukan dengan menguji sensor MPU9250 menggunakan *Kalman Filter* dan tanpa filter yang sudah menggunakan *Digital Motion Processor (DMP)* untuk mengetahui nilai sumbu Y (*Pitch*), nilai sumbu X (*Roll*) dan sumbu Z (*Yaw*).

1. Nilai sumbu Y (*Pitch*)

Berikut ini adalah tabel pengujian dan gambar pada saat pengujian sensor MPU9250 menggunakan gabus yang sudah diukur menggunakan penggaris sudut derajat.

Tabel 3 Tabel Pengujian Nilai Pitch sensor MPU9250

Tabel Pengujian Nilai Pitch sensor MPU9250					
Percobaan Ke-		Parameter Pengujian			
		0°	30°	60°	90°
1	Rata-Rata (°) Kalman Filter	1.221	31.912	62.052	91.922
	Rata-Rata (°) Tanpa Filter	1.220	31.915	62.056	91.929
	Error (°)	0.0115	0.0154	0.0204	0.0115
2	Rata-Rata (°) Kalman Filter	1.198	30.059	61.386	92.258
	Rata-Rata (°) Tanpa Filter	1.197	30.062	61.391	92.268
	Error (°)	0.0157	0.0156	0.0162	0.0107
3	Rata-Rata (°) Kalman Filter	1.173	31.329	61.394	89.579
	Rata-Rata (°) Tanpa Filter	1.173	31.333	61.398	89.572
	Error (°)	0.0149	0.0148	0.0167	0.0144
4	Rata-Rata (°) Kalman Filter	1.212	31.309	62.142	92.160
	Rata-Rata (°) Tanpa Filter	1.213	31.312	62.148	92.166
	Error (°)	0.0155	0.0145	0.0142	0.0112
Total Nilai Error Pengujian Data		0.014575°			

Pengujian yang dilakukan pada sumbu Y (*pitch*) menghasilkan perbedaan nilai data yang tipis berdasarkan parameter pengujian. Dalam empat kali percobaan yang dilakukan dapat diketahui nilai *error* dari sensor MPU9250. Nilai *error* didapatkan berdasarkan hasil regresi linier yang dimana *input Y range* adalah variable terikat yaitu nilai sensor MPU9250 menggunakan *Kalman Filter* dan *input X range* adalah variable bebas yaitu nilai sensor MPU9250 tanpa menggunakan *Filter*.

Dari hasil pengujian yang telah dilakukan diatas, dapat dihitung nilai *error* rata-rata pengujian, berikut adalah perhitungan *error* rata-rata.

$$Error\ Rata - rata = \frac{Jumlah\ banyak\ Error}{Banyaknya\ Error\ terjadi} \quad (1)$$

$$Error\ Rata - rata = \frac{0.0115 + 0.0154 + 0.0204 + 0.0115 + 0.0157 + 0.0156 + 0.0162 + 0.0107 + 0.0149 + 0.0148 + 0.0167 + 0.0144 + 0.0155 + 0.0145 + 0.0142 + 0.0112}{16}$$

$$= 0.014575^\circ$$

Dari hasil pengujian yang telah dilakukan menggunakan sensor MPU9250 pada sumbu Y (*Pitch*), dapat diketahui nilai *error* rata-rata sensor MPU9250 adalah 0.014575°. Nilai yang didapatkan dari sensor dijadikan perbandingan dengan nilai busur derajat untuk mendapatkan nilai sudut yang akurat 100% sehingga dapat digunakan sebagai nilai acuan. Dengan demikian hasil yang didapatkan sensor MPU9250 sudah bekerja dengan baik.

2. Nilai sumbu X (Roll)

Berikut ini adalah tabel pengujian dan gambar pada saat pengujian sensor MPU9250 menggunakan gabus yang sudah diukur menggunakan penggaris sudut derajat.

Tabel 4 Tabel Pengujian Nilai Roll sensor MPU9250

Percobaan Ke-		Tabel Pengujian Nilai Roll Sensor MPU9250			
		Parameter Pengujian			
		0°	30°	60°	90°
1	Rata-Rata (°) Kalman Filter	0.526	30.812	60.44	90.147
	Rata-Rata (°) Tanpa Filter	0.526	30.812	60.441	90.147
	<i>Error</i>	0.0095	0.0117	0.0144	0.0150
2	Rata-Rata (°) Kalman Filter	0.284	31.67	61.127	90.171
	Rata-Rata (°) Tanpa Filter	0.284	31.67	61.128	90.171
	<i>Error</i>	0.0099	0.0111	0.0128	0.0155
3	Rata-Rata (°) Kalman Filter	0.289	30.889	61.171	90.244
	Rata-Rata (°) Tanpa Filter	30.891	30.891	61.173	90.244
	<i>Error</i>	0.0097	0.0112	0.0164	0.0175
4	Rata-Rata (°) Kalman Filter	0.289	30.991	60.938	90.131
	Rata-Rata (°) Tanpa Filter	30.993	30.993	60.939	90.131
	<i>Error</i>	0.0097	0.0109	0.0127	0.0154
Total Nilai <i>Error</i> Pengujian Data		0.012713°			

Pengujian yang dilakukan pada sumbu X (Roll) menghasilkan perbedaan nilai data yang tipis berdasarkan parameter pengujian. Dalam empat kali percobaan yang dilakukan dapat diketahui nilai *error* dari sensor MPU9250. Nilai *error* didapatkan berdasarkan hasil regresi linier yang dimana *input Y range* adalah variable terikat yaitu nilai sensor MPU9250 menggunakan *Kalman Filter* dan *input X range* adalah variable bebas yaitu nilai sensor MPU9250 tanpa menggunakan *Filter*.

Dari hasil pengujian yang telah dilakukan diatas, dapat dihitung nilai *error* rata-rata pengujian, berikut adalah perhitungan *error* rata-rata.

$$Error\ Rata - rata = \frac{Jumlah\ banyak\ Error}{Banyaknya\ Error\ terjadi} \quad (2)$$

$$Error\ Rata - rata = \frac{0.0095 + 0.0117 + 0.0144 + 0.0015 + 0.0099 + 0.0111 + 0.0128 +$$

$$\frac{0.0155 + 0.0097 + 0.0112 + 0.0164 + 0.0175 + 0.0097 + 0.0109 + 0.0127 + 0.0154}{16}$$

$$= 0.012713^\circ$$

Dari hasil pengujian yang telah dilakukan menggunakan sensor MPU9250 pada sumbu X (Roll), dapat diketahui nilai *error* rata-rata sensor MPU9250 adalah 0.012713°. Nilai yang didapatkan dari sensor dijadikan perbandingan dengan nilai busur derajat untuk mendapatkan nilai sudut yang akurat 100% sehingga dapat digunakan sebagai nilai acuan. Dengan demikian hasil yang didapatkan menggunakan sensor MPU9250 bekerja dengan cukup baik.

3. Nilai sumbu Z (*Yaw*)

Berikut ini adalah tabel pengujian dan gambar pada saat pengujian sensor MPU9250 menggunakan penggaris sudut 360° derajat.

Tabel 5 Tabel Pengujian Nilai Yaw Sensor MPU9250

Tabel Pengujian Nilai Yaw Sensor MPU9250					
Percobaan Ke-		Parameter Pengujian			
		0°	30°	60°	90°
1	Rata-Rata (°) Kalman Filter	0.654	29.143	59.595	88.097
	Rata-Rata (°) Tanpa Filter	0.655	29.139	59.590	88.092
	<i>Error</i>	0.0165	0.0075	0.0071	0.0076
2	Rata-Rata (°) Kalman Filter	0.764	29.262	60.046	88.851
	Rata-Rata (°) Tanpa Filter	0.765	29.257	60.041	88.846
	<i>Error</i>	0.0165	0.0075	0.0077	0.0074
3	Rata-Rata (°) Kalman Filter	0.715	29.174	58.893	88.389
	Rata-Rata (°) Tanpa Filter	0.719	29.168	58.888	88.384
	<i>Error</i>	0.0080	0.0070	0.0072	0.0070
4	Rata-Rata (°) Kalman Filter	1.464	29.122	58.976	88.029
	Rata-Rata (°) Tanpa Filter	1.469	29.117	58.972	88.024
	<i>Error</i>	0.0077	0.0068	0.0073	0.0072
Total Nilai <i>Error</i> Pengujian Data		0.008500°			

Pengujian yang dilakukan pada sumbu Z (*Yaw*) menghasilkan perbedaan nilai data berdasarkan parameter pengujian. Dalam empat kali percobaan yang dilakukan dapat diketahui nilai *error* dari sensor MPU9250. Nilai *error* didapatkan berdasarkan hasil regresi linier yang dimana *input Y range* adalah variable terikat yaitu nilai sensor MPU9250 menggunakan *Kalman Filter* dan *input X range* adalah variable bebas yaitu nilai sensor MPU9250 tanpa menggunakan *Filter*.

$$\text{Error Rata - rata} = \frac{\text{Jumlah banyak Error}}{\text{Banyaknya Error terjadi}} \quad (3)$$

$$\text{Error Rata - rata} = \frac{0.0165 + 0.0075 + 0.0071 + 0.0076 + 0.0165 + 0.0075 + 0.0077 +$$

$$\frac{0.0074 + 0.0080 + 0.0070 + 0.0072 + 0.0070 + 0.0077 + 0.0068 + 0.0073 + 0.0072}{16}$$

$$= 0.008500^\circ$$

Dari hasil pengujian yang telah dilakukan menggunakan sensor MPU9250 pada sumbu Z (*Yaw*), dapat diketahui nilai *error* rata-rata sensor MPU9250 adalah 0.008500%. Nilai yang didapatkan dari sensor dijadikan perbandingan dengan nilai busur derajat untuk mendapatkan nilai sudut yang akurat 100% sehingga dapat digunakan sebagai nilai acuan. Dengan demikian nilai *error* yang didapatkan menggunakan sensor MPU9250 mendekati 0% sehingga sensor MPU9250 bekerja dengan cukup baik.

3.2 Hasil Pengujian Delay Waktu sensor MPU9250

Pengujian ini dilakukan dengan menghitung *delay* waktu data sensor saat dikirimkan ke aplikasi *Unity3d*.

Tabel 6 Hasil Pengujian Delay Waktu Sensor MPU9250 Tanpa Filter

Parameter Pengujian	Percobaan Ke-			
	1	2	3	4
1 Menit	0.0075 ms	0.0076 ms	0.0073 ms	0.0109 ms
2 Menit	0.0121 ms	0.0121 ms	0.0097 ms	0.0129 ms
3 Menit	0.0098 ms	0.0123 ms	0.0127 ms	0.0133 ms

Tabel 7 Hasil Pengujian Delay Waktu Sensor MPU9250 Menggunakan Filter

Parameter Pengujian	Percobaan Ke-			
	1	2	3	4
1 Menit	0.0088 ms	0.0076 ms	0.0070 ms	0.0086 ms
2 Menit	0.0108 ms	0.0103 ms	0.0106 ms	0.0134 ms
3 Menit	0.0090 ms	0.0106 ms	0.0110 ms	0.0110 ms

Pada Tabel 6 dan Tabel 7 di atas dapat dilihat rata-rata waktu *delay* untuk setiap data sensor MPU9250 menggunakan *Kalman Filter* dan data sensor MPU9250 tanpa *filter* memiliki waktu *delay* yang berbeda-beda yang dipengaruhi oleh kecepatan koneksi wi-fi yang digunakan.

3.3 Hasil Pengujian Kesesuaian Gerakan Sensor MPU9250

Berikut merupakan hasil presentase kesesuaian gerakan sensor MPU9250 yang akan menggerakkan player pada aplikasi Unity3d.

Tabel 8 Hasil Pengujian Kesesuaian Gerak Sensor Menggunakan filter

Pengujian	Heading	Sarung tangan ke Atas	Sarung tangan ke Bawah	Sarung tangan ke Kanan	Sarung tangan ke kiri	%
1	Depan	✓	✓	✓	✓	100
	Kanan	✓	✓	✓	✓	100
	Kiri	✓	✓	✓	✓	100
	Belakang	✓	✓	✓	✓	100
2	Depan	✓	✓	✓	✓	100
	Kanan	✓	✓	✓	✓	100
	Kiri	✓	✓	✓	✓	100
	Belakang	✓	✓	✓	✓	100
3	Depan	✓	✓	✓	✓	100
	Kanan	✓	✓	✓	✓	100
	Kiri	✓	✓	✓	✓	100
	Belakang	✓	✓	✓	✓	100
4	Depan	✓	✓	✓	✓	100
	Kanan	✓	✓	✓	✓	100
	Kiri	✓	✓	✓	✓	100
	Belakang	✓	✓	✓	✓	100
5	Depan	✓	✓	✓	✓	100
	Kanan	✓	✓	✓	✓	100
	Kiri	✓	✓	✓	✓	100
	Belakang	✓	✓	✓	✓	100
Rata - Rata						100

Pada Tabel 8 di atas dapat dilihat data sensor MPU9250 tanpa menggunakan filter dan data sensor MPU9250 menggunakan *Kalman Filter* memberikan presentase 100% sesuai dengan gerakan sensor MPU9250 terhadap karakter di dalam *Unity3D*.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini adalah sebagai berikut :

1. Berdasarkan hasil pengujian, prototipe sarung tangan dengan sensor MPU9250 berhasil dibuat dan berfungsi dengan baik dengan total nilai *error* sebesar 0.012713° pada pengujian sumbu X (*Roll*), sedangkan pada pengujian sumbu Y (*Pitch*) memiliki total nilai *error* sebesar 0.014575° dan pada sumbu Z (*Yaw*) memiliki total nilai *error* 0.008500° .
2. Pada pengujian parameter waktu pengiriman data sensor MPU9250 yang dilakukan sebanyak 4 kali percobaan. Menghasilkan waktu pengiriman data tercepat sebesar 0.0070 ms dan waktu pengiriman data terlama sebesar 0.0134 ms.
3. Pada kinerja keseluruhan sistem prototipe sarung tangan untuk kesesuaian gerakan sensor MPU9250 dengan karakter yang ada pada *Unity3D* memperoleh 100% kesuksesan dalam kesesuaian gerak.

4.2 Saran

Adapun saran yang diberikan untuk penelitian selanjutnya adalah sebagai berikut :

1. Menggunakan sensor yang lebih terkini agar mendapat akurasi data yang lebih baik lagi.
2. Menggunakan perangkat wifi yang standar lebih baru agar mendapatkan waktu pengiriman data yang lebih baik lagi.

DAFTAR PUSTAKA

- [1] M. P. I Wayan Artanayasa, S.Pd., 2014. "*Panahan*". Graha Ilmu Cetakan Pertama.
- [2] B. Sihite, F. Samopa, and N. A. Sani, "Pembuatan Aplikasi 3D Viewer Mobile dengan Menggunakan Teknologi Virtual Reality (Studi Kasus: Perobekan Bendera Belanda di Hotel Majapahit)," *Tek. Pomits*, vol. 2, no. 2, pp. 397–400, 2013.
- [3] Akbar Hanief, "Struktur Sebagai Ekspresionisme Arsitektur Nusantara Pada Sarana Olahraga Sasaran di Kota Baru Parahyangan," *Jurnal Reka Karsa* pp. 1–6, 2016.
- [4] Suryajaya, I Dewa Bagas, "TEKNIK MOTION CAPTURE DALAM PROSES PEMBUATAN ANIMASI 3D MENGGUNAKAN MICROSOFT KINECT," Seminar Nasional Teknologi Informasi dan Multimedia pp. 6–8, 2015.
- [5] M. Nasrulloh, "EKSPRESI WAJAH NON-HUMAN MODEL 3D BERBASIS RADIAL BASIS FUNCTION MENGGUNAKAN DATA FACIAL MOTION CAPTURE FACIAL EXPRESSIONS NON-HUMAN 3D MODEL BASED RADIAL BASIS FUNCTIONS USING FACIAL MOTION," 2015.
- [6] Alma'i, Vidi Rahman, Wahyudi, S.T, M.T dan Iwan Setiawan, S.T, M.T. "Aplikasi Sensor *Accelerometer* Pada Deteksi Posisi," 2011.
- [7] I. Fahruzi and E. S. Abdullah, "Integrasi Sensor Multifungsi *Accelerometer* untuk Mendeteksi Kekuatan Benturan," *KNSI*, 2014.
- [8] M. I. Nugraha *et al.*, "ANALISIS PENGARUH MOMENT GYROSCOPE PADA KESIMBANGAN PENDULUM CARTESSIAN," *Jurnal Manutech* Vol. 9. no 2, pp. 1–7, 2017.
- [9] Bahri and R. Primadianti, "Pemantauan Pergerakan Orang Lanjut Usia Secara Real-Time," *J. Institusi USU*, pp. 243–253, 2017.
- [10] A. Aziz, "Implementation Markerless Augmented Reality Using Android Sensors For Identification of Buildings in Sebelas Maret University," vol. 5, no. 1, pp. 25–33, 2016.
- [11] A. Larasati, R. V. H. Ginardi, and S. Sarwosri, "Aplikasi Scoring Perilaku Pengemudi Menggunakan Sensor Fusi," *J. Tek. ITS*, vol. 7, no. 2, 2018.
- [12] Bangkin Harry, Mamat Ruhimat. "Kalibrasi Magnetometer Tipe 1540 Menggunakan Kalibrator Magnetometer," *Buletin Komdrad* Vol.4, no 2, 2012.
- [13] A. U. Darajat, M. Komarudin, and S. R. S, "SISTEM TELEMETRI *Unmanned Aerial Vehicle* (UAV) BERBASIS *Inertial Measurement Unit* (IMU)," *ELECTRICAN Jurnal Rekayasa dan Tekonologi Elektro* no. 3, pp. 169–177, 2012.
- [14] Suryanti, Desta Ika, "INERTIAL MEASUREMENT UNIT (IMU) PADA SISTEM PENGENDALI SATELIT," *Faktualita* vol. 12, no. 2, pp. 7–9, 2017.
- [15] S. Yuliani and H. M. Saputra, "Kolaborasi Kalman Filter dengan Complementary Filter untuk Mengoptimasi Hasil Sensor Gyroscope dan Accelerometer Kolaborasi Kalman Filter dengan Complementary Filter untuk Mengoptimasi Hasil Sensor Gyroscope dan Accelerometer," *Itenas*, 2017.
- [16] Somari, Florus Herman. "SISTEM DATA LOGGER PERALATAN ELEKTRONIK BERBASIS ANDROID," 2017.
- [17] Sasmoko Dani, Yunar Arief Wicaksono. "IMPLEMENTASI PENERAPAN INTERNET of THINGS (IoT) PADA MONITORING INFUS MENGGUNAKAN ESP 8266 DAN WEB UNTUK BERBAGI DATA." *Jurnal Ilmiah Informasi* Vol. 2, no 1, 2017.
- [18] Mahfudz, "Aplikasi Augmented Reality Pengenalan Alat Musik Gamelan Berbasis Android," 2019.