

## LAMPIRAN V

### PENGEMBANGAN PERANGKAT KERAS KONTROL BUSUR DAN ANAK PANAH BERBASIS SENSOR IMU DENGAN METODE COMPLEMENTARY FILTER

#### *DEVELOPMENT OF HARDWARE CONTROL BOW AND ARROW BASED ON IMU SENSOR USING COMPLEMENTARY FILTER METHOD*

Fery Chandra<sup>#1</sup>, Farel Mohammad Fauzi<sup>#2</sup>

<sup>1,2</sup>Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom  
ferychandra@student.telkomuniversity.ac.id, farelft@telkomuniversity.ac.id

#### Abstrak

Dalam kehidupan sehari-hari, manusia sering menggunakan tangan untuk saling berinteraksi antara manusia dengan mesin, ataupun manusia dengan komputer. Dalam game panahan, panahan adalah suatu kegiatan menggunakan busur untuk menembakkan anak panah. *Unity* merupakan suatu aplikasi untuk mengembangkan game *multiplatform* yang didesain untuk lebih mudah digunakan. Dalam penelitian ini penulis membuat suatu alat yaitu prototipe kontrol busur dan anak panah berbasis sensor imu dengan menggunakan metode *Complementary Filter*, yang dimana kedua tangan dapat digunakan sebagai *controller* dengan menggunakan sensor imu MPU 9250 dan NodeMCU ESP 8266 yang dimana akan di implementasikan ke dalam game *archery* menggunakan *unity*.

**Kata kunci :** Panahan, *Unity*, *Complementary Filter*, MPU9250, NodeMCU ESP 8266.

#### Abstract

In daily life, humans often use their hands to interact between humans and machines, or humans and computers. In archery games, archery is an activity using a bow to shoot arrows. *Unity* is an application for developing multi-platform games that are designed to be easier to use. In this study the authors made a tool that is a prototype of arc sensor and arrow-based sensor using the *Complementary Filter* method, where both hands can be used as a controller using the IMU sensor 9250 MPU and NodeMCU ESP 8266 which will be implemented in the archery game by using the *unity*.

**Keywords:** Archery, *Unity*, *Complementary Filter*, MPU9250, NodeMCU ESP 8266.

#### 1. Pendahuluan

Panahan merupakan permainan yang banyak diminati oleh semua orang, dan merupakan permainan yang menggunakan busur untuk melepaskan anak panah. Permainan panahan membutuhkan kekuatan pada tubuh seperti kedua otot tangan, otot dada, bahu dan punggung. Pada saat melakukan permainan panahan dibutuhkan konsentrasi yang penuh untuk memanah. Permainan panahan adalah salah satu permainan yang membutuhkan busur dan anak panah untuk pengaplikasiannya, dengan menembakkan anak panah menggunakan busur panah untuk mengenai sasaran dengan jarak yang telah ditentukan[1].

*Virtual Reality (VR)* merupakan sebuah inovasi teknologi terbaru yang dikembangkan untuk memungkinkan seseorang melakukan suatu interaksi secara nyata terhadap objek grafis dengan visualisasi 3D atau gambar berbentuk hologram [2]. Teknologi yang sangat canggih ini mampu memberikan sebuah pengalaman baru serta menyenangkan bagi penggunaannya, karena pengguna seolah-olah bisa menyentuh objek tersebut secara langsung melalui *virtual reality*.

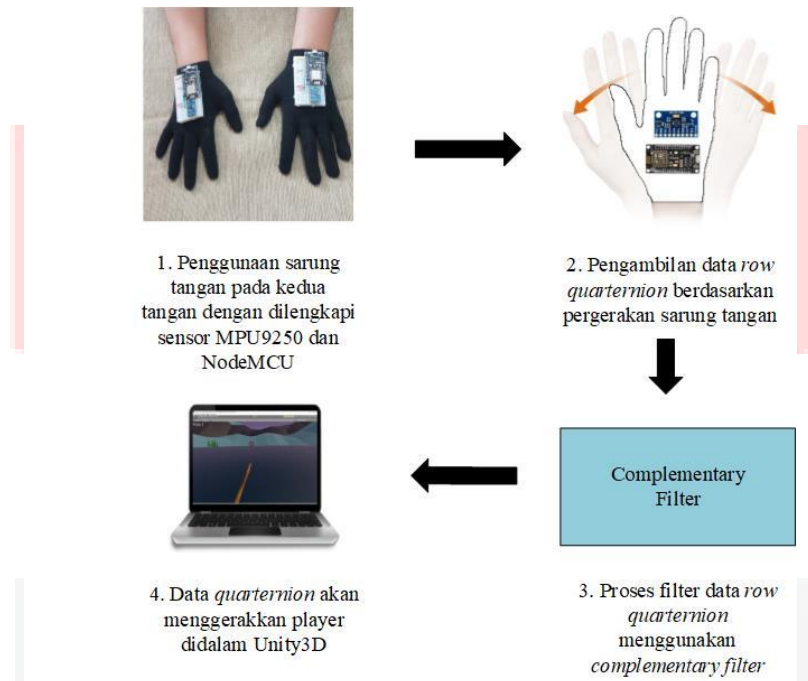
Terdapat permasalahan yang membuat permainan panahan di Indonesia mengalami hambatan pada perkembangan yaitu, karena permainan panahan tersebut membutuhkan tempat yang luas untuk melakukan perlombaan maupun untuk latihan. Karena permainan panahan memiliki beberapa kategori dalam jarak tembak ke target yaitu 18 meter sampai dengan 90 meter dengan lebar 4,5 meter sampai 5 meter, yang dimana sangat sulit untuk menemukan tempat dengan panjang dan lebar lapangan yang dibutuhkan di daerah perkotaan[3].

Berdasarkan permasalahan yang ada diatas, dibutuhkan solusi untuk menyelesaikan permasalahan tersebut yaitu dengan membuat perangkat keras kontrol busur dan anak panah berbasis sensor IMU dengan menggunakan metode algoritma *Complementary Filter*. Kontrol busur dan anak panah berbasis sensor IMU ini akan di implementasikan ke dalam *Unity* menggunakan *virtual reality (VR)* sehingga tidak membutuhkan tempat yang luas untuk melakukan kegiatan panahan.

## 2. Perancangan Sistem

### 2.1 Gambaran Umum Sistem

Sistem yang dibuat berdasarkan pergerakan kedua tangan pengguna, yang akan menggerakkan player dan melepaskan anak panah didalam game panahan pada *unity* menggunakan metode *Complementary Filter*. Simulasi sistem ini dibuat dengan tujuan untuk memainkan game panahan menggunakan prototipe sarungan tangan yang menggunakan sensor MPU9250. Berikut gambaran umum sistem pada gambar 1



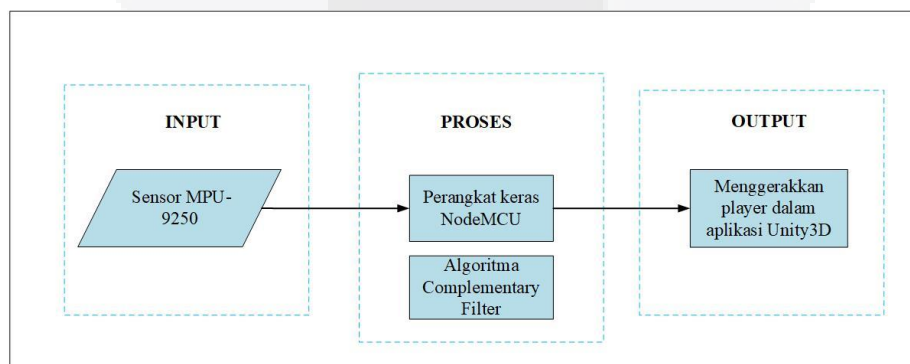
Gambar 1. Gambaran umum sistem

Pada gambar 1 menggambarkan tentang gambaran umum dari prototipe sarung tangan :

1. Sarung tangan akan digunakan pada kedua tangan seperti pada gambar di atas.
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 *Complementary Filter*.
4. Data setelah di filter akan dikirimkan kedalam *Unity* untuk menggerakkan player dan melepaskan anak panah pada game panahan.

### 2.2 Analisis Kebutuhan Sistem

Pada analisis sistem prototipe sarung tangan ini membahas tentang 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 *Complementary 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 menunjukkan 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 diakses 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 menunjukkan 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 dan 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. Hasil Pengujian Nilai Pitch

Percobaan Ke-		Parameter Pengujian			
		0 <sup>0</sup>	30 <sup>0</sup>	60 <sup>0</sup>	90 <sup>0</sup>
1	Rata-Rata (°) <i>Complementary Filter</i>	0,454	33,396	61,631	90,542
	Rata-Rata (°) Tanpa Filter	0,474	33,405	61,644	90,530
	<i>Error</i> (°)	0,022706	0,032879	0,028749	0,031802
2	Rata-Rata (°) <i>Complementary Filter</i>	0,360	31,920	60,665	90,685
	Rata-Rata (°) Tanpa Filter	0,368	30,784	60,671	90,713
	<i>Error</i> (°)	0,009066	0	0,040661	0,082449
3	Rata-Rata (°) <i>Complementary Filter</i>	0,287	31,886	60,935	90,703
	Rata-Rata (°) Tanpa Filter	0,287	31,920	60,935	90,685
	<i>Error</i> (°)	3,34E-17	0,170927	6,01E-16	0,05191
4	Rata-Rata (°) <i>Complementary Filter</i>	0,195	31,063	60,650	90,685
	Rata-Rata (°) Tanpa Filter	0,195	31,063	60,652	90,713
	<i>Error</i> (°)	2,1E-17	6,33E-16	0,087596	0,082687
Total Nilai <i>Error</i>		0,084804 derajat			

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 *Complementary 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.

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

$$\begin{aligned} Error\ Rata - rata &= \frac{0,022706 + 0,032879 + 0,028749 + 0,031802 + 0,009066 + 0 + 0,040661 + 0,082449 + \\ &= \frac{3,34E - 17 + 0,170927 + 6,01E - 16 + 0,05191 + 2,1E - 17 + 6,33E - 16 + 0,087596 + 0,82687}{16} \\ &= 0,084804\ derajat \end{aligned}$$

Dari hasil pengujian yang telah dilakukan menggunakan sensor MPU9250 pada sumbu Y (*pitch*), dapat diketahui nilai *error* rata-rata sensor MPU9250 adalah 0,084804 derajat. 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. Hasil Pengujian Nilai Roll

Percobaan Ke-		Parameter Pengujian			
		0°	30°	60°	90°
1	Rata-Rata (°) <i>Complementary Filter</i>	0,816	32,744	61,982	98,574
	Rata-Rata (°) Tanpa Filter	0,816	31,277	62,786	98,574
	<i>Error</i> (°)	0,000812	0,047336	0,66441	0,00213
2	Rata-Rata (°) <i>Complementary Filter</i>	0,444	31,401	62,596	98,615
	Rata-Rata (°) Tanpa Filter	0,816	30,952	61,596	98,615
	<i>Error</i> (°)	0,044761	0,367777	1,811779	0,004101
3	Rata-Rata (°) <i>Complementary Filter</i>	0,825	31,111	63,828	97,562
	Rata-Rata (°) Tanpa Filter	0,828	31,414	62,238	97,562
	<i>Error</i> (°)	0,050133	0,114547	2,220078	5,81E-15
4	Rata-Rata (°) <i>Complementary Filter</i>	0,511	31,783	64,293	98,559
	Rata-Rata (°) Tanpa Filter	0,463	31,563	62,222	98,559
	<i>Error</i> (°)	0,046046	0,044098	2,12854	1,73E-16
Total Nilai <i>Error</i>		0,471659 derajat			

Pengujian yang dilakukan pada sumbu Y (*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 *Complementary 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.

$$Error\ Rata-rata = \frac{Jumlah\ banyak\ Error}{Banyaknya\ Error\ terjadi} \tag{2}$$

$$Error\ Rata - rata = \frac{0,000812 + 0,047336 + 0,66441 + 0,00213 + 0,044761 + 0,367777 + 1,811779 + 0,004101 + 0,050133 + 0,114547 + 2,220078 + 5,81E - 15 + 0,046046 + 0,044098 + 2,12854 + 1,73E - 16}{16}$$

= 0,471659 derajat

Dari hasil pengujian yang telah dilakukan menggunakan sensor MPU9250 pada sumbu Y (*pitch*), dapat diketahui nilai *error* rata-rata sensor MPU9250 adalah 0,084804 derajat. 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.

3. Nilai sumbu Z (*Yaw*)

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

Tabel 5. Hasil Pengujian Nilai Yaw

Tabel Pengujian Nilai Yaw Sensor MPU 9250					
Percobaan Ke-		Parameter Pengujian			
		0°	30°	60°	90°
1	Rata-Rata (°) <i>Complementary Filter</i>	0,492	30,682	60,615	90,858
	Rata-Rata (°) Tanpa Filter	0,534	30,682	60,615	90,706
	<i>Error</i> (°)	0,099521	3,97E-15	0,002053	0,000848
2	Rata-Rata (°) <i>Complementary Filter</i>	0,624	30,769	60,370	90,862
	Rata-Rata (°) Tanpa Filter	0,624	30,769	60,370	90,858
	<i>Error</i> (°)	0	8,1E-16	1,24E-16	0,022287
3	Rata-Rata (°) <i>Complementary Filter</i>	1,015	30,762	60,418	90,771
	Rata-Rata (°) Tanpa Filter	1,020	30,762	60,420	90,773
	<i>Error</i> (°)	0,014399	1,36E-15	0,012494	0,01495
4	Rata-Rata (°) <i>Complementary Filter</i>	0,298	30,369	60,417	90,628
	Rata-Rata (°) Tanpa Filter	0,298	30,369	60,424	90,612
	<i>Error</i> (°)	3,8E-16	4,56E-16	0,06494	0,037388
Total Nilai <i>Error</i> Pengujian Data		0,016805 derajat			

Pengujian yang dilakukan pada sumbu Y (*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 *Complementary 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.

$$Error \text{ Rata-rata} = \frac{\text{Jumlah banyak Error}}{\text{Banyaknya Error terjadi}}$$

(3)

*Error Rata – rata*

$$= \frac{0,099521 + 3,97E - 15 + 0,002053 + 0,000848 + 0 + 8,1E - 16 + 1,24E - 16 + 0,022287 + 0,014399 + 1,36E - 15 + 0,012494 + 0,01495 + 3,8E - 16 + 4,56E - 16 + 0,06494 + 0,037388}{16}$$

$$= 0,016805 \text{ derajat}$$

Dari hasil pengujian yang telah dilakukan menggunakan sensor MPU9250 pada sumbu Y (*pitch*), dapat diketahui nilai *error* rata-rata sensor MPU9250 adalah 0,084804 derajat. 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.

### 3.2 Hasil Pengujian Waktu Pengiriman Data Sensor IMU

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

Tabel 6. Hasil Pengujian *Delay* Waktu Sensor IMU Menggunakan *Complementary Filter*

Parameter Pengujian	Percobaan Ke -			
	1	2	3	4
1 menit	0,018106674 ms	0,009713807 ms	0,009694070 ms	0,010084343 ms
2 menit	0,007802098 ms	0,008832748 ms	0,009344596 ms	0,009613812 ms
3 menit	0,008840998 ms	0,008860491 ms	0,008536473 ms	0,008676138 ms

Pada **Tabel 4.** diatas dapat dilihat rata-rata waktu *delay* untuk setiap data sensor IMU DMP menggunakan *Complementary Filter* dan data sensor IMU DMP tanpa *filter* memiliki waktu *delay* yang berbeda – beda yang dipengaruhi oleh kecepatan koneksi *wi-fi* yang digunakan.

### 3.3 Hasil Pengujian Kesesuaian Gerakan Sensor Menggunakan Filter

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

Tabel 7 Hasil Pengujian Kesesuaian Gerak Sensor Menggunakan *Complementary 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



#### 4. Kesimpulan

Kesimpulan yang di peroleh dari penelitian ini adalah sebagai berikut:

1. Berdasarkan dari hasil pengujian, simulasi perangkat keras sarung tangan kontrol busur dan anak panah berbasis sensor IMU, data sensor IMU DMP menggunakan *Complementary Filter* dan tidak menggunakan *filter* menghasilkan nilai *error pitch*, *roll* dan *yaw*. Nilai *error pitch* diperoleh sebesar 0,084804 derajat, untuk nilai *error roll* diperoleh sebesar 0,471659 derajat dan nilai *error yaw* diperoleh sebesar 0,016805 derajat.
2. Pada pengujian parameter waktu delay pengiriman data sensor IMU yang dilakukan sebanyak 4 kali percobaan dengan menggunakan DMP dan *Complementary Filter* menghasilkan lama waktu terkecil yaitu 0,007802098 ms dan waktu delay yang terbesar diperoleh 0,018106674 ms.
3. Pada kinerja keseluruhan sistem prototipe sarung tangan untuk kesesuaian gerakan sensor IMU dengan player yang ada pada *Unity3D* memperoleh 100% kesuksesan dalam kesesuaian gerak.

#### 5. Saran

Adapun saran yang diberikan untuk penelitian selanjutnya adalah sebagai berikut:

1. Pemilihan *Hardware* seperti sensor IMU MPU9250 harus di perhatikan agar *support* dengan bahasa pemrograman yang akan digunakan.
2. Menggunakan perangkat *wifi* yang standar lebih baru agar mendapatkan waktu pengiriman data yang lebih baik.

#### Daftar Pustaka:

- [1] I Wayan Artanayasa, S.Pd., M.pd. 2014. "Panahan" Graha Ilmu Cetakan Pertama.
- [2] Singh, N., & Singh, S. (2017). "Virtual reality", A brief survey. 2017 International Conference on Information Communication and Embedded Systems (ICICES).
- [3] Hanief Akbar. 2016. "Struktur Sebagai Ekspresionisme Arsitektur Nusantara Pada Sarana Olahraga Sasaran di Kota Baru Parahyangan," *Jurnal Online Institut Teknologi Nasional*.
- [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. Furniss, "Motion Capture," Retrieved May,1 2010 From <http://web.mit.edu/commforum/papers/furniss.html>, 1999. (diakses pada tanggal 25 September 2019).
- [7] M. Kitakawa and B. Windsor, "MoCap for Artists: workflow and techniques for motion capture," Massachusetts, USA.: Elsevier, 2008.
- [8] S. Dyer, J. Martin, and J. Zulauf, "Motion Capture White Paper," Retrieved May,25 2013
- [9] Gunawan, Ade Putra. (2012). Robot Keseimbangan Menggunakan Pengendali PID (Software).
- [10] 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.
- [11] 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.
- [12] Mpu-9250 nine-axis (gyro + accelerometer + compass) mems motiontracking device. [Online]. Available: <http://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/> (diakses pada tanggal 25 September 2019).
- [13] Walter T Higgins, JR, 1975 "A Comparison of Complementary and Kalman Filtering", IEEE Transactions on Aerospace and Electronic Systems, Vol. AES- 11, No.3.
- [14] Saputra, L. K. P., & Lukito, Y. (2017). Implementation of air conditioning control system using REST protocol based on NodeMCU ESP8266.
- [15] R. K. Kodali and S. Soratkal, 2016. "MQTT based home automation system using ESP8266," 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), 2016.