

Rancang Bangun Platform Gerak Untuk Simulator Pesawat Udara: Perancangan Pemrograman Arduino

Yusuf Salim¹, Muhammad Adam Abhinaya², Pajar Firmansyah³, Singgih Satrio Wibowo⁴

¹Jurusan Teknik Mesin, Politeknik Negeri Bandung, Bandung 40012

¹E-mail : yusuf.salim.aer21@polban.ac.id

²E-mail : muhammad.adam.aer21@polban.ac.id

³E-mail : pajar.firmansyah.aer21@polban.ac.id

⁴E-mail : singgih.wibowo@polban.ac.id

ABSTRAK

Penelitian ini bertujuan untuk merancang dan membangun sebuah platform simulasi gerak derajat kebebasan (*Degrees of Freedom*) untuk menciptakan pergerakan pesawat udara dengan menggunakan *motor wiper* sebagai dasarnya. Platform gerak ini dikendalikan menggunakan oleh Arduino dan diprogram dengan kebutuhannya. Metode penyelesaian masalah yang digunakan meliputi studi literatur, menentukan alat, Arduino dan perancangan dan pembuatan pemrograman, serta pengujian atau percobaan dari hasil program yang dirancang dan evaluasi. Platform simulasi gerak yang dibangun mampu memberikan gerakan yang realistis dan akurat berdasarkan parameter yang ditentukan. Melalui penggunaan *motor wiper*, platform ini dapat memberikan kontrol yang baik terhadap simulasi gerakan. Selain itu, penggunaan Arduino dan pemrograman yang tepat memungkinkan pengaturan dan kontrol yang efisien. Hasil pengujian menunjukkan bahwa platform ini memiliki tingkat keberhasilan yang memuaskan dalam merancang dan membangun simulasi gerak pesawat udara. Secara singkat, platform simulasi gerak ini dapat digunakan dalam berbagai aplikasi seperti penelitian, hiburan serta pelatihan di bidang yang membutuhkan simulasi gerak yang akurat dan realistis.

Kata Kunci

Derajat kebebasan, pergerakan pesawat udara, Arduino, dan pemrograman

ABSTRACT

This research aims to design and build a degrees of freedom movement simulation platform to create aircraft movements using a wiper motor as the basis. This motion platform is controlled using Arduino and programmed to suit its needs. The problem-solving methods used include literature study, tool determination, Arduino and programming design and creation, as well as testing or experimenting with the results of the designed program and evaluation. The motion simulation platform built is capable of providing realistic and accurate movements based on specified parameters. Through the use of a wiper motor, this platform can provide fine control over simulated movements. Additionally, the use of Arduino and proper programming allows for efficient setup and control. Test results show that this platform has a satisfactory level of success in designing and building aircraft motion simulations. In short, this motion simulation platform can be used in various applications such as research, entertainment and training in fields that require accurate and realistic motion simulation.

Keywords

Degrees of freedom, aircraft movement, Arduino, and programming

1. PENDAHULUAN

Teknologi simulasi menjadi semakin focus utama di berbagai aplikasi seperti pelatihan, penelitian, dan hiburan virtual. Untuk menciptakan

pengalaman simulasi yang menyerupai kenyataan atau realitas. Aspek penting adalah platform gerak simulasi dengan derajat kebebasan yang cukup untuk pergerakan pesawat udara, yaitu *rolling*, *pithcing*, dan *yawing*. Dalam hal ini, penggunaan *motor wiper* sebagai dasar penggerak menjadi

pilihan yang menarik karena dapat memberikan keandalan dan ketersediaan yang cukup. (1)(2)

Platform simulasi gerak derajat kebebasan (*Degrees of Freedom*) dengan semakin banyaknya derajat kebebasan, maka akan semakin mahal dan rumit untuk mekanismenya. Untuk penelitian ini dengan menggunakan 3 derajat kebebasan untuk menghasilkan pergerakan pesawat, yaitu *rolling*, *pitching*, dan *yawing* yang memungkinkan simulasi pergerakan benda atau kendaraan dalam tiga dimensi, menghasilkan pengalaman yang lebih mendalam dan menarik. (2)

Pemanfaatan Arduino sebagai platform pengembangan perangkat keras (*hardware*) terbuka membuka peluang untuk meningkatkan kontrol dan pemrograman dalam implementasi platform simulasi gerak. Dengan menggunakan sistem kendali PID (*Proportional Integral Derivative*) untuk menciptakan pergerakan yang akurat. Oleh karena itu, tujuan utama penelitian ini adalah untuk mengkaji secara mendalam terhadap penggunaan Arduino dan teknik pemrograman yang sesuai untuk merancang dan membangun platform gerak 3 derajat kebebasan dengan dasar *motor wiper*. (3)

Dengan pemanfaatan Arduino dan penerapan teknik pemrograman yang sesuai, platform gerak ini diharapkan dapat memberikan akurasi dan daya tanggap yang tinggi. Selain itu, diharapkan dapat berkontribusi terhadap pengembangan teknologi simulasi canggih yang semakin terjangkau. Pemahaman menyeluruh tentang interaksi antara Arduino, *motor wiper*, dan pemrograman sangat penting untuk meningkatkan kualitas dan kegunaan platform simulasi gerak untuk berbagai aplikasi. (4)

2. TINJAUAN PUSTAKA

Siam, M. F. Mohd dan dkk dalam Journal of the Society of Automotive Engineers Malaysia yang berjudul "*Driving Simulator Development with Two Degrees of Freedom Motion for Driver Behavior Study*" menghasilkan Arduino sebagai pengontrol dalam pembuatan simulasi gerakan dengan komponen salah satunya yaitu *motor DC*, dan menghasilkan gerakan 2 Degrees of Freedom (DOF) yaitu *roll* dan *pitch*. Program tersebut terhubung dengan perangkat keras (*hardware*) yaitu *steering wheel* yang digunakan sebagai alat pengendalinya. (1)

Pouliot, Nicolas A. dan Clement M. Gosselin dalam Journal of Aircraft yang berjudul "*Motion Simulation Capabilities of Three-Degree-of-*

Freedom Flight Simulators" menghasilkan analisis perbandingan antara simulator tiga derajat kebebasan mampu menghasilkan kualitas simulasi gerak yang sebanding dengan yang dihasilkan oleh platform Stewart enam derajat kebebasan. (2)

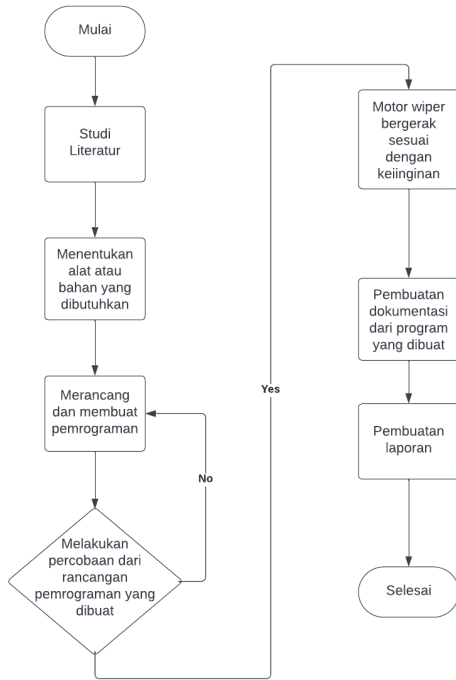
Wibowo, Singgih Satrio dalam paper yang berjudul "*Algoritma Kinematika Balik Platform Stewart Dengan Servo Putar*" berisikan tentang penurunan matematis pada kasus kinematika balik (*inverse kinematic*) dan juga penggunaan pengontrol PID (*Proportional-Integral-Derivative Controller*) dalam pembuatan perangkat simulator bergerak 2 DOF (*Degrees of Freedom*). (3)

Recktenwald, Gerald W. dan David E. Hall dalam ASEE Annual Conference and Exposition, Conference Proceedings yang berjudul "*Using Arduino as a Platform for Programming, Design and Measurement in a Freshman Engineering Course*" berisikan tentang arduino sebagai platform untuk pemrograman. (4)

Penelitian ini penulis akan melakukan rancang bangun platform gerak simulasi pesawat udara untuk dapat berbagai pengaplikasian salah satunya hiburan virtual. Penelitian ini dikerjakan bersama dengan rekan yang bernama Pajar Firmansyah pengkajian terhadap struktur yang dibuat dan Muhammad Adam Abinayah pengkajian terhadap elektriknya.

3. METODE PENYELESAIAN MASALAH

Metode penyelesaian masalah yang digunakan untuk menyelesaikan penelitian ini dilakukan secara sistematis, dimulai dari studi literatur sampai pengetasan apakah codingannya berjalan atau tidak. Metode penyelesaian masalah digambarkan pada *flowchart* di bawah ini:



Gambar 1. Flowchart metode penyelesaian masalah

3.1 Studi Literatur

Studi literatur dilakukan untuk memahami Arduino, mulai dari alatnya dan juga aplikasi untuk pemrogramannya, serta memahami untuk pemrogramannya. Adapun sumber yang dipilih untuk menyelesaikan penelitian berasal dari jurnal-jurnal yang sesuai dengan judul yang dipilih dan juga *paper*, selain itu ada dari buku yang khusus membahas Arduino yang berjudul “Belajar Cepat dan Pemrograman Arduino”.

3.2 DRO

Design, Recruitment, Object (DRO) merupakan hal yang penting dalam perancangan. Karena DRO ini menjadi patokan untuk spesifikasi yang ingin dibuat. DRO pada penelitian, yaitu perancangan dalam pembuatan gerak simulator. Adapun rincian DRO yang dapat dilihat pada tabel 1.

Tabel 1. DRO perancangan gerak simulator.

No.	Desain Requirement	Objectives
1.	Degrees of Freedom	3 (rolling, pitching, yawing) (Coupled Movement)
2.	Platform Rolling Capability	+10° dan -10°
2.	Platform Pitching Capability	+10° dan -5°
3.	Platform Yawing Capability	+10° dan -10°
4.	Maximum load	120 kg

5.	Frame Material	Hollow Steel	Galvanic
6.	Power	24V	
7.	Mikrokontroler	Arduino UNO dan Arduino Mega 2560	
8.	Motor DC	Motor Wiper	
9.	Motor Driver Modul	BTS7960	43A High Power
10.	Programming Language	C	

3.3 Alat atau Bahan

Adapun alat/bahan yang dibutuhkan untuk penunjang-penunjang menyelesaikan penelitian ini ditunjukkan pada tabel 2.

Tabel 2. List alat atau bahan.

No.	Nama Alat	Jumlah
1.	Arduino UNO R3	3
2.	Arduino Mega 2560	1
3.	BTS7960 43A High Power Motor Driver Module	3

3.4 Pembuatan Pemrograman

Perancangan dan pembuatan pemrograman dilakukan pada aplikasi atau *software* Arduino IDE versi 2.2.1 yang dimana nantinya akan di aplikasikannya ke Arduino UNO R3 (sebagai *slave* untuk menerima perintah dari master) dan Arduino Mega 2560 (sebagai master untuk mengirimkan perintah ke *slave*) agar dapat mengendalikan *motor wiper*nya dengan menggunakan inputan data dari serial monitor pada master serta potentiometer sebagai sensor untuk menghentikan pergerakan *motor wiper*.

Pada tabel 3, pin yang digunakan adalah pin komunikasi serial RX1 dan TX1 untuk komunikasi serial 1, pin komunikasi serial RX2 dan TX2 untuk komunikasi serial 2, dan pin komunikasi serial RX3 dan TX3 untuk komunikasi serial 3.

Tabel 3. Kodingan untuk Arduino sebagai master.

```

1: // =====
2: // Program untuk mengendalikan 3 Motor Wiper
3: // Author : Yusuf Salim <yusufsalim023@gmail.com>
4: // History : Start in 2024-03-25
5: // =====
6: // Referensi
7: // [1] Paper - 2024 - Algoritma Kinematika Balik
8: // Platform Stewart Dengan Servo Putar,
9: // 20240309, Bab 1, 2 dan 5 Singgih Satrio
10: // Wibowo - COMP 4.docx
11: // [2] https://docs.arduino.cc/learn/built-in
12: // libraries/software-serial/ (20240616)
13:
14: // Master
15:
16: const int initialSetpoint1 = 7200; // Nilai setpoint
17: // awal untuk
18: // Slave 1
19: const int initialSetpoint2 = 7200; // Nilai setpoint
20: // awal untuk
21: // Slave 2
22: const int initialSetpoint3 = 7200; // Nilai setpoint
23: // awal untuk
24: // Slave 3
25: int setpoint1 = initialSetpoint1; // Variabel untuk
26: // menyimpan
27: // setpoint awal
28: // pada Slave 1

```

```

29: int setpoint2 = initialSetpoint2; // Variabel untuk
30: // menyimpan
31: // setpoint awal
32: // pada Slave 2
33: int setpoint3 = initialSetpoint3; // Variabel untuk
34: // menyimpan
35: // setpoint awal
36: // pada Slave 3
37:
38: void setup() {
39:   Serial.begin(115200); // Inisialisasi komunikasi
40:   // serial untuk debugging
41:   Serial1.begin(115200); // Inisialisasi komunikasi
42:   // serial perangkat lunak
43:   // dengan slave 1
44:   Serial2.begin(115200); // Inisialisasi komunikasi
45:   // serial perangkat lunak
46:   // dengan slave 2
47:   Serial3.begin(115200); // Inisialisasi komunikasi
48:   // serial perangkat lunak
49:   // dengan slave 3
50:   Serial.println("Masukkan setpoint dengan format:
51: <Setpoint1> <Setpoint2> <Setpoint3>");
52:   Serial.println("Contoh: 360 240 180 untuk Slave 1, Slave 2,
53: dan Slave 3");
54:
55:   // Kirim nilai setpoint awal ke slave
56:   Serial1.println(setpoint1);
57:   delay(50);
58:   Serial2.println(setpoint2);
59:   delay(50);
60:   Serial3.println(setpoint3);
61:   delay(50);
62: } Akhir dari void setup()
63:
64: void loop() {
65:   // Baca input dari serial monitor jika tersedia
66:   if (Serial.available() > 0) {
67:     String input = Serial.readStringUntil('\n'); // Baca input
68:     // dari
69:     // serial
70:     // monitor
71:     input.trim(); // Menghapus spasi di awal dan akhir
72:     Serial.print("Input received: ");
73:     Serial.println(input); // Tampilkan input yang diterima
74:
75:     // serial monitor
76:
77:     int firstSpaceIndex = input.indexOf(' ');
78:     int secondSpaceIndex = input.indexOf(' ', firstSpaceIndex
79: + 1);
80:
81:     if (firstSpaceIndex > 0 && secondSpaceIndex >
82: firstSpaceIndex) {
83:       int newSetpoint1 = input.substring(0,
84: firstSpaceIndex).toInt();
85:       int newSetpoint2 = input.substring(firstSpaceIndex + 1,
86: secondSpaceIndex).toInt();

```

```

87:       int newSetpoint3 = input.substring(secondSpaceIndex +
88: 1).toInt();
89:
90:       setpoint1 = newSetpoint1;
91:       setpoint2 = newSetpoint2;
92:       setpoint3 = newSetpoint3;
93:
94:       Serial.print("Setpoint for Slave 1 updated to: ");
95:       Serial.println(setpoint1); // Tampilkan nilai setpoint
96:       // yang baru di serial
97:
98:       monitor
99:       delay(100);
100:      Serial1.println(setpoint1); // Kirim nilai setpoint ke
101:      // Slave 1
102:      delay(100);
103:
104:      Serial.print("Setpoint for Slave 2 updated to: ");
105:      Serial.println(setpoint2); // Tampilkan nilai setpoint
106:      // yang baru di serial
107:
108:      monitor
109:      delay(100);
110:      Serial2.println(setpoint2); // Kirim nilai setpoint ke
111:      // Slave 2
112:      delay(50);
113:
114:      Serial.print("Setpoint for Slave 3 updated to: ");
115:      Serial.println(setpoint3); // Tampilkan nilai setpoint
116:      // yang baru di serial
117:
118:      monitor
119:      Serial3.println(setpoint3); // Kirim nilai setpoint ke
120:      // Slave 3
121:      } else {
122:        Serial.println("Setpoint tidak valid. Format harus
123: <Setpoint1> <Setpoint2> <Setpoint3>");
124:      }
125:      delay(100); // Tambahkan sedikit delay untuk memastikan
126:      data selesai dikirim
127:      // Tambahkan delay untuk menghindari flooding pada
128:      komunikasi serial
129:      delay(100);
130:    } Akhir dari void loop()

```

Pada tabel 4, pin yang digunakan adalah pin digital, yaitu pin digital 5, 6, 7, 8 untuk BTS7960 Motor Driver, pin analog A0 untuk Potentiometer, dan pin RX0 dan TX0 untuk komunikasi serialnya dengan cara di silang dengan master pada pin RX1TX1, RX2TX2, dan RX3TX3.

Tabel 4. Kodingan untuk Arduino sebagai slave.

```

1: //
2: =====
3: // Author YS, 2024-06-16
4: // based on work by SSW and others
5: //
6: =====
7: // Referensi
8: // [1] Paper - 2024 - Algoritma Kinematika Balik
9: // Platform Stewart Dengan Servo Putar,
10: // 20240309, Bab 1, 2 dan 5 - // Singgih
11: // Satrio Wibowo - COMP 1.docx
12: // [2] https://docs.arduino.cc/learn/built-in
13: // libraries/software-serial/ (20240616)
14:
15: // Slave 1
16: #include <SoftwareSerial.h> // Memanggil library
17: // SoftwareSerial
18: // Pin untuk BTS7960
19: #define RPWM 7
20: #define LPWM 8
21: #define R_EN 5
22: #define L_EN 6
23:
24: // Pin untuk Potentiometer
25: #define POT_PIN A0
26:
27: SoftwareSerial Serial1(0, 1); // RX, TX
28: // Komunikasi
29: Serial 1
30: // Sebagai Penerima
31:
32: int val = 0;
33: String receivedData;
34: int i_roll_d;
35: double d_roll_d;
36:
37: float kp = 0.5;
38: float ki = 0.0;
39: float kd = 0.00;
40:
41: float Theta, Theta_d;
42: int dt;

```

```

43: float K_Theta = 1.424; // Konstanta Pulley
44:
45: unsigned long t;
46: unsigned long t_prev = 0;
47:
48: float e, e_prev = 0, inte, inte_prev = 0, e_der;
49: float Vmax = 12;
50: float Vmin = -12;
51: float V = 0.1;
52:
53: void WriteDriverVoltage(float V, float Vmax) {
54:   int PWMval = int(255 * abs(V) / Vmax);
55:   if (PWMval > 255) {
56:     PWMval = 255;
57:   }
58:   if (V > 0) {
59:     digitalWrite(RPWM, HIGH);
60:     digitalWrite(LPWM, LOW);
61:   } else if (V < 0) {
62:     digitalWrite(RPWM, LOW);
63:     digitalWrite(LPWM, HIGH);
64:   } else {
65:     digitalWrite(RPWM, LOW);
66:     digitalWrite(LPWM, LOW);
67:   }
68:   analogWrite(R_EN, PWMval);
69:   analogWrite(L_EN, PWMval);
70: } // Akhir dari void WriteDriverVoltage
71:
72: void setup() {
73:   pinMode(RPWM, OUTPUT);
74:   pinMode(LPWM, OUTPUT);
75:   pinMode(R_EN, OUTPUT);
76:   pinMode(L_EN, OUTPUT);
77:   pinMode(POT_PIN, INPUT);
78:   digitalWrite(R_EN, HIGH);
79:   digitalWrite(L_EN, HIGH);
80:   Serial.begin(115200);
81:   Serial1.begin(115200);
82: } // Akhir dari void setup()
83:
84: void loop() {

```

```

85:   val = analogRead(POT_PIN); // Pembacaan
86:       // Potensiometer
87:   Theta = val * K_Theta; // Nilai potensiometer
88:   yang
89:       // sudah dikalikan
90:   dengan
91:       // konstanta pulley
92:
93:   // Menerima input nilai dari Master
94:   if (Serial1.available() > 0) {
95:       // Membaca data yang diterima dari Serial1
96:       receivedData = Serial1.readStringUntil('\n');
97:       receivedData.trim(); // Menghapus spasi di
98:       // sekitar string
99:       // (jika ada)
100:
101:       // Debug: Cetak data yang diterima untuk
102:       // memeriksa
103:       Serial.print("Received raw data: ");
104:       Serial.println(receivedData);
105:
106:       // Konversi string ke integer
107:       i_roll_d = receivedData.toInt();
108:
109:       // Pastikan konversi berhasil (jika
110:       diperlukan)
111:       if (i_roll_d == 0 && receivedData != "0") {
112:           Serial.println("Error: Failed to convert
113: string to integer!");
114:           // Lakukan penanganan kesalahan di sini
115:       jika
116:           // diperlukan
117:       } else {
118:           // Skala data jika diperlukan
119:           d_roll_d = i_roll_d / 10.0;
120:           Theta_d = d_roll_d;
121:
122:           // Gunakan nilai Theta_d untuk tujuan
123:           // selanjutnya
124:       }
125:
126:
127:       // Proses kontrol PID
128:       t = millis();
129:       dt = (t - t_prev);
130:       e = Theta_d - Theta;
131:       inte = inte_prev + (dt * (e + e_prev) / 2);
132:       e_der = (e - e_prev) / dt;
133:       V = kp * e + ki * inte + kd * e_der;
134:
135:       if (V > Vmax) {
136:           V = Vmax;
137:       }
138:       if (V < Vmin) {
139:           V = Vmin;
140:       }
141:
142:       WriteDriverVoltage(V, Vmax);
143:
144:       t_prev = t;
145:       inte_prev = inte;
146:       e_prev = e;
147:
148:       // Debug Output
149:       Serial.print("Theta_d: ");
150:       Serial.print(Theta_d);
151:       Serial.print("\t");
152:       Serial.print("POT: ");
153:       Serial.println(Theta);
154:   } // Akhir dari void loop()

```

4. HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian

Hasil pengujian dilakukan terhadap pada pemrograman yang dibuat terhadap *platform*.

Pada tabel 3 sebagai master dalam Arduino, diharuskan menginput nilai pada serial monitor agar dapat memerintah kepada *slave* untuk bergerak sesuai dengan nilai yang diinput. Nilai potensiometer pada *slave* di mulai dari posisi 5x putaran atau setara dengan nilai pada serial monitor adalah 720. Maka data awal sebelum

diinput adalah 7200. Data yang diinput dari master diharuskan x10 dari data yang diterima oleh *slave*, misalnya input nilai 7200 maka data yang diterima oleh *slave* adalah 720. Pada tabel 5 merupakan data yang harus diinput agar platform melakukan pergerakan, seperti *rolling*, *pitching*, dan *yawing*.

Tabel 5. Nilai yang diinput dari master

	Pergerakan					
	Platform Rolling		Platform Pitching		Platform Yawing	
	+10°	-10°	+10°	-5°	+10°	-10°
Serial1	6300	8100	6300	8100	-	-
Serial2	6300	8100	8100	6300	-	-
Serial3	-	-	-	-	6300	8100

Pada tabel 6-8 adalah nilai yang diterima pada *slave*.

Tabel 6. Nilai yang diterima pada slave 1

	Pergerakan			
	Platform Rolling		Platform Pitching	
	+10°	-	+10°	-5°
			10°	
Nilai yang diterima	630	810	630	810
Nilai pada Potentiometer	630	810	630	810

Tabel 7. Nilai yang diterima pada slave 2

	Pergerakan			
	Platform Rolling		Platform Pitching	
	+10°	-	+10°	-5°
			10°	
Nilai yang diterima	630	810	810	630
Nilai pada Potentiometer	630	810	810	630

Tabel 8. Nilai yang diterima pada slave 3

	Pergerakan Platform Yawing	
	+10°	-10°
Nilai yang diterima	630	810
Nilai pada Potentiometer	630	810

4.2 Pembahasan

Hasil pengujian pada tabel 5-8 tidak terdapat perbedaan pada nilai potentiometer dengan nilai data yang diterima pada *slave* dari data yang diinput pada *master*. Pada posisi awal, potentiometer diputar sebanyak 5x atau setara dengan nilai 720 pada serial monitor di *slave* dan tanduk *motor wiper* diposisikan sejajar dengan *platform* dan dengan posisi *motor wiper* 1 dan 2 berbanding terbalik. Posisi *motor wiper* 1 ada di sebelah kiri *platform*, posisi *motor wiper* 2 ada di sebelah kanan *platform*, dan posisi *motor wiper* 3 ada di belakang *platform*, sebagaimana pada gambar 2 dan gambar 3.



Gambar 2. Posisi tanduk motor wiper 1 dan 2



Gambar 3. Posisi tanduk motor wiper 3

Untuk melakukan pergerakan *platform rolling +10°*, untuk tanduk *motor wiper* 2 melakukan pergerakan ke arah bawah sebanyak 90° dan tanduk *motor wiper* 1 pergerakan ke arah atas sebanyak 90° dan kedua *motor wiper* akan bergerak secara *clockwise*. Maka dari itu, dikarenakan melakukan gerakan sebanyak 90° dan searah nilai yang diinput untuk *slave* 1 dan *slave* 2 adalah sama dan nilai input dari serial monitor diharuskan ditambah 900 dari nilai awal, menjadi 8100. Pada *slave* 1 dan *slave* 2 tidak terjadinya error dengan data yang diterima dan nilai potentiometer sama dengan nilai data yang diterima. Pada gambar 4 merupakan pergerakan *platform rolling +10°*.

potentiometer sama dengan nilai data yang diterima. Pada gambar 4 merupakan pergerakan *platform rolling +10°*.



Gambar 4. Platform rolling $+10^\circ$

Untuk melakukan pergerakan *platform rolling -10°*, untuk tanduk *motor wiper* 2 melakukan pergerakan ke arah atas sebanyak 90° dan tanduk *motor wiper* 1 pergerakan ke arah bawah sebanyak 90° dan kedua *motor wiper* akan bergerak secara *counter clockwise*. Maka dari itu, dikarenakan melakukan gerakan sebanyak 90° dan searah nilai yang diinput untuk *slave* 1 dan *slave* 2 adalah sama dan nilai input dari serial monitor diharuskan ditambah 900 dari nilai awal, menjadi 8100. Pada *slave* 1 dan *slave* 2 tidak terjadinya error dengan data yang diterima dan nilai potentiometer sama dengan nilai data yang diterima. Pada gambar 5 merupakan pergerakan *platform rolling -10°*.



Gambar 5. Platform rolling -10°

Untuk melakukan pergerakan *platform pitching* $+10^\circ$, untuk tanduk *motor wiper 2* melakukan pergerakan ke arah atas sebanyak 90° dan tanduk *motor wiper 1* pergerakan ke arah atas sebanyak 90° dan kedua *motor wiper* akan bergerak secara berlawanan, *motor wiper 1* akan bergerak *clockwise* dan *motor wiper 2* akan bergerak *counter clockwise*. Maka dari itu, dikarenakan melakukan gerakan sebanyak 90° dan berlawanan, nilai yang diinput untuk *slave 1* adalah nilai input dari serial monitor diharuskan dikurang 900 dari nilai awal, menjadi 6300 dan nilai yang diinput untuk *slave 2* adalah nilai input dari serial monitor diharuskan ditambah 900 dari nilai awal, menjadi 8100. Pada *slave 1* dan *slave 2* tidak terjadinya error dengan data yang diterima dan nilai potentiometer sama dengan nilai data yang diterima. Pada gambar 6 merupakan pergerakan *platform pitching* $+10^\circ$.



Gambar 6. Platform pitching $+10^\circ$

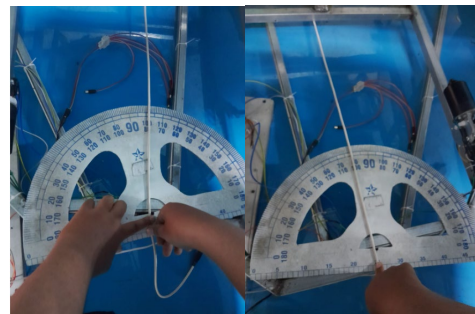
Untuk melakukan pergerakan *platform pitching* -5° , untuk tanduk *motor wiper 2* melakukan pergerakan ke arah bawah sebanyak 90° dan tanduk *motor wiper 1* pergerakan ke arah bawah sebanyak 90° dan kedua *motor wiper* akan bergerak secara berlawanan, *motor wiper 1* akan bergerak *counter clockwise* dan *motor wiper 2* akan bergerak *clockwise*. Maka dari itu, dikarenakan melakukan gerakan sebanyak 90° dan berlawanan, nilai yang diinput untuk *slave 1* adalah nilai input dari serial monitor diharuskan ditambah 900 dari nilai awal, menjadi 8100 dan nilai yang diinput untuk *slave 2* adalah nilai input dari serial monitor diharuskan dikurang 900 dari nilai awal, menjadi 6300. Pada *slave 1* dan *slave 2*

tidak terjadinya error dengan data yang diterima dan nilai potentiometer sama dengan nilai data yang diterima. Pada gambar 7 merupakan pergerakan *platform pitching* -5° .



Gambar 7. Platform pitching -5°

Dilihat pada tabel 8, nilai yang diterima oleh *slave 3* dengan data yang diinput dari master tidak terdapat perbedaan, serta nilai potentiometer sama dengan nilai yang diterima. Akan tetapi, *platform* tidak dapat bergerak, dikarenakan *joint* yang terhubung dengan tanduk *motor wiper 3* terlalu tegang. Karena hal ini yang menyebabkan tanduk *motor wiper 3* tidak dapat menggerakkan *platform*. Pada gambar 8 merupakan hasil pengukuran dengan menggunakan busur derajat dengan menggerakkan *platform* secara manual untuk pergerakan *platform yawing* $+10^\circ$ dan -10° .



Gambar 8. Pengukuran menggunakan busur derajat untuk platform yawing $+10^\circ$ dan -10°

Dari hasil pembahasan tersebut, dilihat data yang asli pada *slave* di serial monitor, terdapat perbedaan sedikit pada nilai potentiometer dengan nilai yang diterima. Hal ini dikarenakan *motor wiper* menerima beban dari *platform* sehingga *motor wiper* tersebut mempertahankan posisi

sesuai dengan data yang diterima dan karena hal tersebut tidak di anggap sebagai error.

Pergerakan *motor wiper* dan pergerakan tanduk *motor wiper* sudah sesuai dengan data yang diinput. Akan tetapi, untuk pergerakan *rolling* dan *pitching* mengalami pergerakan *yawing*. Hal ini disebabkan oleh *joint* yang terhubung di tengah *platform* merupakan *joint* yang dapat bergerak bebas.

5. KESIMPULAN

Dari perancangan dan hasil pemrograman yang dibuat, maka dapat disimpulkan, yaitu:

1. Pada pemrograman yang dibuat, berhasil melakukan serial komunikasi dari master (Arduino Mega 2560) terhadap 3 *slave* (Arduino Uno) yang dimana ketika mengirinkan data terhadap *slave* dari master, *slave* tersebut berhasil menerima data dengan baik, sehingga *motor wiper* bergerak mencapai sensor (potentiometer) yang sesuai dengan data yang di input.
2. Mengacu pada pembahasan pemrograman terhadap *platform*, penelitian ini dinyatakan belum sepenuhnya berhasil. Dikarenakan untuk pergerakan *yawing* dengan menggunakan *motor wiper* tidak dapat bergerak. Hal ini disebabkan *joint* pada *motor wiper* 3 terlalu kaku, sehingga *motor wiper* 3 tidak dapat menggerakkan *platform*.

DAFTAR PUSTAKA

- [1] Siam M. F. M., Borhan N., Sukardi A. Driving simulator development with two degrees of freedom motion for driver behavior study. *Journal of the Society of Automotive Engineers Malaysia*. January 2017;1(1):4-11. DOI: 10.56381/jsaem.v1i1.2
- [2] Pouliot N. A., Gosselin C. M., Nahon M. A. Motion simulation capabilities of three-degree-of-freedom flight simulators. *Journal of Aircraft*. January - February 1998;35(1):9-17. DOI: 10.2514/2.2283.
- [3] Wibowo S. S. Algoritma kinematika balik platform stewart dengan servo putar. Paper Penelitian. Unpublished.
- [4] Recktenwald G. W., Hall D. E. Using arduino as a platform for programming, design and measurement in a freshman engineering course. *ASEE Annual Conference and Exposition, Conference Proceedings*. DOI: 10.18260/1-2--18720.
- [5] Sheldon R. Degrees of freedom (mechanics). [internet]. September 2022. [Di akses 1 Mei 2024]. Tersedia dari: <https://www.techtargget.com/whatis/definition/n/degrees-of-freedom>.
- [6] Andrianto H., Darmawan A. Belajar cepat dan pemrograman arduino. Ed ke-2. Bandung: Informatika Bandung; 2017.
- [7] Prastyo E. A. Arduino uno: Pengertian dan spesifikasinya. [internet]. Februari 2023. [Di akses 1 Mei 2024]. Tersedia dari: <https://www.arduino.biz.id/2023/02/arduino-uno-pengertian-dan-spesifikasinya.html>.
- [8] Handson Technology. Bts7960 high current 43a h-bridge motor driver. [internet]. Tanpa Tahun. [Online]. [Di akses 1 Mei 2024]. Tersedia dari: <https://www.handsontec.com/dataspeccs/modul/e/BTS7960%20Motor%20Driver.pdf>.
- [9] Saragih R. R. Pemrograman dan bahasa pemrograman. [internet]. June 2018. [Diakses 1 Mei 2024]. Tersedia dari: [researchgate.net/publication/329885312_PEMROGRAMAN_DAN_BAHASA_PEMROGRAMAN](https://www.researchgate.net/publication/329885312_PEMROGRAMAN_DAN_BAHASA_PEMROGRAMAN).
- [10] Van Heerden A. S. J., Lidbetter R. T., Liebenberg L., Mathews E. H., Meyer J. P. Development of a motion platform for an educational flight simulator. *International Journal of Mechanical Engineering Education*. October 2011; 39(4):306-322. DOI: 10.7227/IJMEE.39.4.4.
- [11] Arthaya B. M., Christian R., Tamba T. A., Tukul D. B. Design and kinematic analysis of a two-dof moving platform as a base for a car simulator. *Journal of Mechatronics, Electrical Power, and Vehicular Technology*. July 2022;13(1):48-59. DOI: 10.14203/j.mev.2022.v13.48-59
- [12] Wu P. L., Chang Y. H., Liao C. S., Chieng W. H. Design for a 2-dof motion platform. *Journal of Robotics and Mechatronics*. February 2011;23(1):19-33. DOI: 10.20965/jrm.2011.p0019
- [13] Moss A., Schluntz R., Buhr P. A. CV: Adding modern programming language features to c. *Computer Science, Engineering*. August 2018;48:2111-2146. DOI: 10.1002/spe.2624
- [14] Montironi M. A., Qian B., Cheng H. H. Development and application of the charduino toolkit for teaching how to program arduino boards through the c/c++ interpreter. *Computer Science, Education, Engineering*. July 2017;1-13. DOI: 10.1002/cae.21854
- [15] Purdum J. Beginning c for arduino. *Computer Science, Engineering*. 2015. DOI: 10.1007/978-1-4842-0940-0