

DAFTAR PUSTAKA

- [1] S. K. Deb, J. H. Rokky, T. C. Mallick, and J. Shetara, "Design and construction of an underwater robot," 2018, doi: 10.1109/ICAEE.2017.8255367.
- [2] L. G. García-Valdovinos, T. Salgado-Jiménez, M. Bandala-Sánchez, L. Nava-Balanzar, R. Hernández-Alvarado, and J. A. Cruz-Ledesma, "Modelling, Design and Robust Control of a Remotely Operated Underwater Vehicle," *Int. J. Adv. Robot. Syst.*, vol. 11, no. 1, 2014, doi: 10.5772/56810.
- [3] S. F. Masoomi *et al.*, "Design and Construction of a Specialised Biomimetic Robot in Multiple Swimming Gaits," *Int. J. Adv. Robot. Syst.*, 2015, doi: 10.5772/60547.
- [4] H. Huang, L. Wan, W. T. Chang, Y. J. Pang, and S. Q. Jiang, "A fault-tolerable control scheme for an open-frame underwater vehicle," *Int. J. Adv. Robot. Syst.*, 2014, doi: 10.5772/58578.
- [5] J. He, Y. Li, Y. Li, Y. Jiang, and L. An, "Fault diagnosis in autonomous underwater vehicle propeller in the transition stage based on GP-RPF," *Int. J. Adv. Robot. Syst.*, vol. 15, no. 6, pp. 1–9, 2018, doi: 10.1177/1729881418814683.
- [6] R. Xu, G. Tang, De Xie, D. Huang, and L. Han, "Underactuated tracking control of underwater vehicles using control moment gyros," *Int. J. Adv. Robot. Syst.*, 2018, doi: 10.1177/1729881417750759.
- [7] L. A. LA Gonzalez, "Design , Modelling and Control of an Autonomous Underwater Vehicle," *BE Thesis, Univ. West. Aust.*, 2004.
- [8] G. Schillaci, F. Schillaci, and V. V. Hafner, "A Customisable Underwater Robot," 2017, [Online]. Available: <http://arxiv.org/abs/1707.06564>.
- [9] R. A. Armstrong, O. Pizarro, and C. Roman, "Underwater Robotic Technology for Imaging Mesophotic Coral Ecosystems," 2019.
- [10] N. Englebert, P. Bongaerts, P. Muir, K. B. Hay, and O. Hoegh-Guldberg, "Deepest zooxanthellate corals of the Great Barrier Reef and Coral Sea,"

Mar. Biodivers., 2015, doi: 10.1007/s12526-014-0221-8.

- [11] H. Anam, J. D. Setiawan, and A. Budiyo, “Simulation and Dynamics Analysis of Remotely Operated Vehicle (ROV) Using PID Controller For Pitch Movement,” *Mar. Underw. Sci. Technol.*, vol. 1, no. 1, pp. 23–31, 2015.

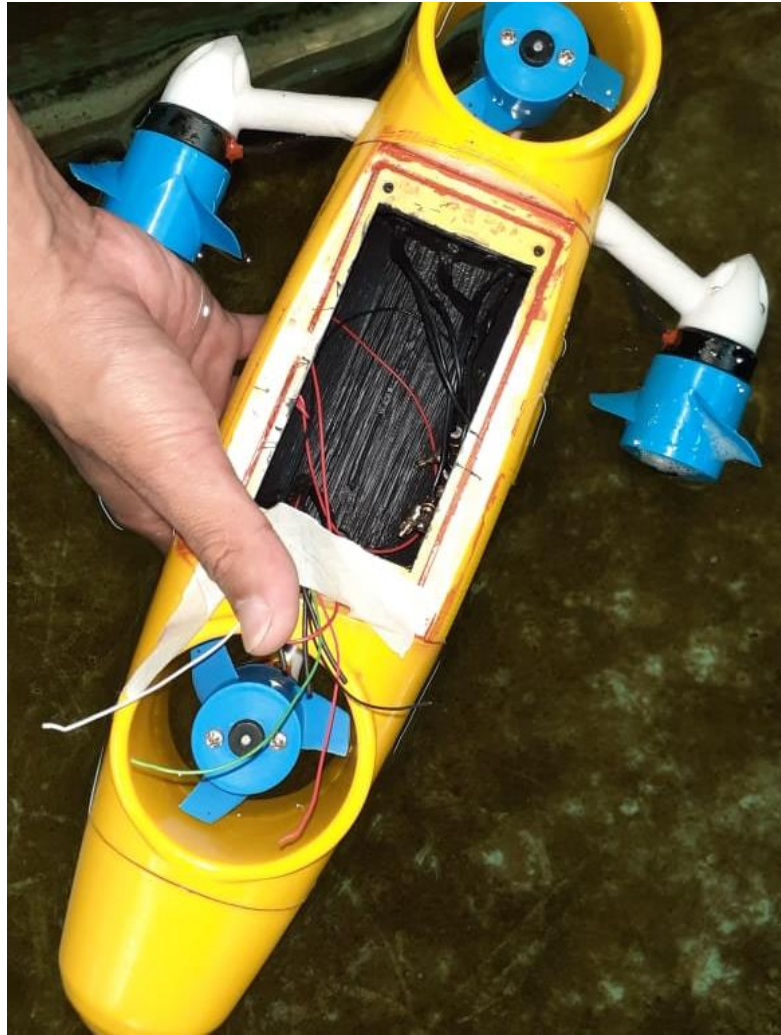
LAMPIRAN



a. Mengukur berat keseluruhan *underwater robot*



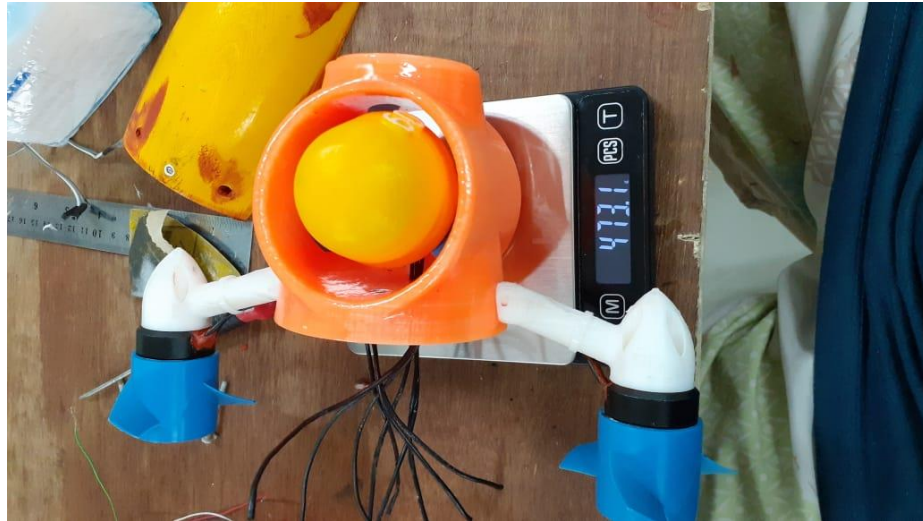
b. Membuat kode program perintah



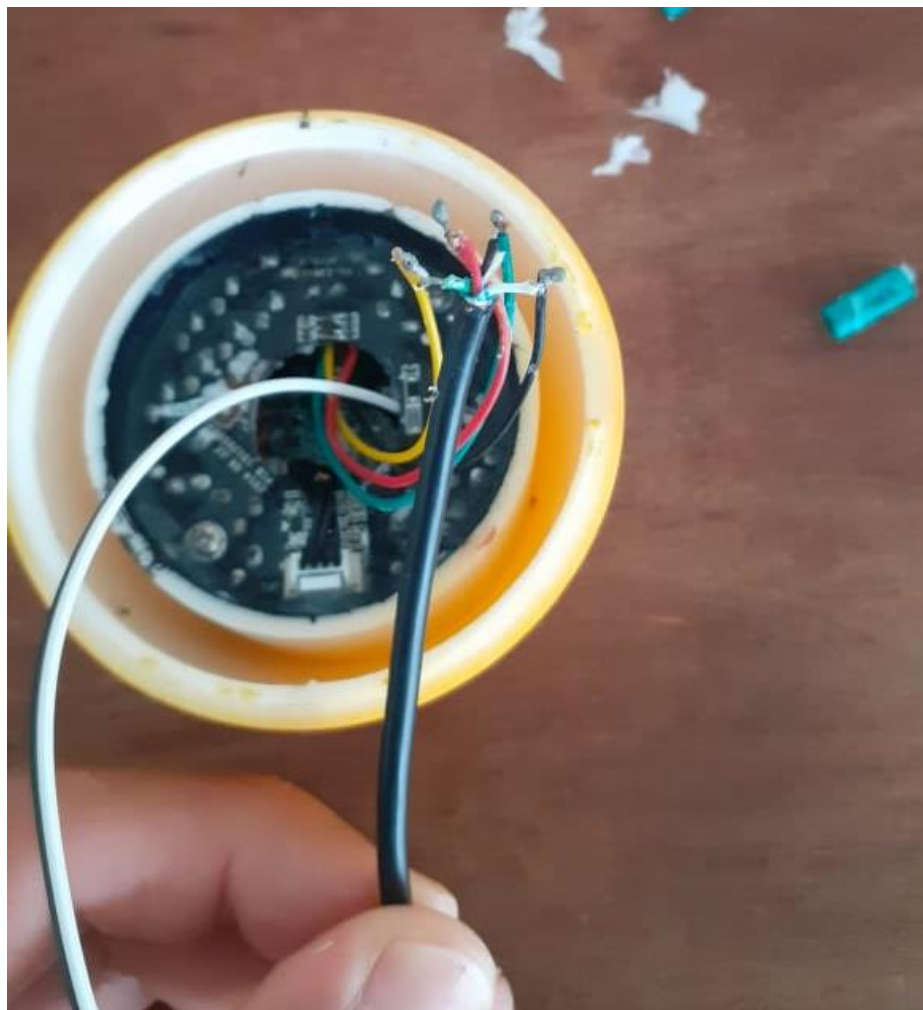
c. Cek kebocoran ruang elektrik



d. Proses merakit part body



e. Mengukur berat total per masing-masing part body



f. Pemasangan module kamera pada body depan



g. Proses cetak part body menggunakan printer 3D

```

#include <SoftwareSerial.h>

//LAMPU
#include <stdio.h>
#include <stdlib.h>

const int rxpin = 3;
const int txpin = 4;

SoftwareSerial serial_ard(rxpin, txpin);
const int PCRjoypin = 0;
const int ThbJoyXpin = 1;
const int ThbJoyYpin = 2;

int bataskiri = 88;
int bataskanan = 97;

int Zval;
int Yval;
int Xval;

int Z,X;
int ZM;
int M2,M4;

const int button = 2;           // GPIO 8 for the button
const int led =12;             // GPIO 7 for the LED
int ledflag=0;                 // LED status flag

void setup(){
Serial.begin(9600);
serial_ard.begin(9600);

pinMode(button,INPUT);        // define button as an input
pinMode(led,OUTPUT);         // define LED as an output
digitalWrite(led,LOW);
}

void loop(){

Zval = analogRead(PCRjoypin);
Zval = map(Zval, 111, 913, 0, 180);

Yval = analogRead(ThbJoyXpin);
Yval = map(Yval, 934, 101, 0, 180);

```

```

Xval = analogRead(ThbJoyYpin);
Xval = map(Xval, 161, 900, 0, 180);

if(Xval >= bataskiri && Xval <= bataskanan){ //TITIK NOL
THROTTLE X
  X=0;
  M2=93;
  M4=93;
}

if(Yval >= bataskiri && Yval <= bataskanan){ //TITIK NOL
THROTTLE Y
  Yval=93;
}

if(Zval >= bataskiri && Zval <= bataskanan){ //TITIK NOL
THROTTLE Z
  Z=0;
  ZM=93;
  M2=93;
  M4=93;
}

if(Zval < bataskiri){ // THROTTLE MUNDUR
Z=map(Zval, bataskiri, 0, 0, 70);
ZM=93+Z;
M2=ZM;
M4=ZM;
}

if(Zval > bataskanan){ // THORTTLE MAJU
Z=map(Zval, bataskanan, 170, 0, -70);
ZM=93+Z;
M2=ZM;
M4=ZM;
}

else{
  if(Xval > bataskanan){ // THROTTLE BELOK KANAN
  X=map(Xval, bataskanan, 180, 0, 50);
  M2=bataskanan+X;
  M4=bataskanan-X;
  }

  if(Xval < bataskiri){ // THROTTLE BELOK KIRI

```



```

        X=map(Xval, bataskiri, 0, 0, -50);
        M2=bataskiri+X;
        M4=bataskiri-X;
    }
}

if (digitalRead(button)==HIGH){ // if button is pressed
    if (ledflag==0) {           // and the status flag is LOW
        ledflag=1;             // make status flag HIGH
        digitalWrite(led,HIGH); // and turn on the LED
    }                           //
    else {                      // otherwise...
        ledflag=0;             // make status flag LOW
        digitalWrite(led,LOW); // and turn off the LED
    }
    Serial.print(ledflag);
    delay(500);
}

serial_ard.print(M2); //Send the PC Raider sensor value to the
Slave Arduino
serial_ard.print(",");
serial_ard.print(M4); //Send the Thumbjoystick X-axis value
value to the Slave Arduino
serial_ard.print(",");
serial_ard.print(Yval); //Send the Thumbjoystick Y-axis value
value to the Slave Arduino
serial_ard.print(","); // final comma to aid message end detection
serial_ard.print(ledflag);
serial_ard.print(",");
serial_ard.println(); // finish the message with a cr/lf.

Serial.print("M2 = ");
Serial.print(M2);
Serial.print(" M4 = ");
Serial.print(M4);
//Serial.print(" X = ");
//Serial.print(x);
//Serial.print(" XM = ");
//Serial.print(XM);
Serial.print(" Z: ");
Serial.print(Zval);
Serial.print(", ");
Serial.print("Y: ");
Serial.print(Yval);
Serial.print(", ");

```

```
Serial.print("X: ");  
Serial.print(Xval);  
Serial.println();  
delay(15);
```

```
}
```

h. Kode program remote kontrol

```
#include <Servo.h>  
  
#include <stdio.h>  
#include <stdlib.h>  
  
Servo M1;  
Servo M2;  
Servo M3;  
Servo M4;  
  
int test;  
  
const int NoF = 4;  
int val[NoF];  
int fieldIndex = 0;  
  
const int led1 = 12;  
int ledflag=0;  
  
void setup()  
{  
Serial.begin(9600);  
Serial1.begin(9600);  
M1.attach(2);  
M2.attach(3);  
M3.attach(4);  
M4.attach(5);  
  
val[0] = val[1] = val[2] = 90;  
test = 90;  
  
pinMode(led1,OUTPUT);      // define LED as an output  
digitalWrite(led1,LOW);  
}  
  
void loop()  
{  
DoStuff();
```

```

for(int i=0;i<4;i++)
{
  Serial.print(val[i]);
  Serial.print(" ");
}
Serial.println();
M1.write(val[2]);
M2.write(val[0]);
M3.write(val[2]);
M4.write(val[1]);
digitalWrite(led1, (val[3]));
delay(15);
}

void DoStuff()
{
  if(Serial1.available())
  {
    for(fieldIndex = 0; fieldIndex < 4; fieldIndex ++)
    {
      val[fieldIndex]=Serial1.parseInt();
    }

    Serial.println();
    fieldIndex = 0;
  }
}

```

i. Kode program *underwater* robot