

## LAMPIRAN A

### Program Arduino IDE

```
#define BLYNK_TEMPLATE_ID "TMPL69oUqqOu5"  
#define BLYNK_TEMPLATE_NAME "rc ship"  
#define BLYNK_AUTH_TOKEN  
"FhCacjkZsHPGPqrbELRXhkJMV9WIAMav"  
  
// #define BLYNK_PRINT Serial // definisi print serial nek blink ws  
konek  
  
#include <WiFi.h> //library esp32 diseting nggo wifi Akses  
point  
#include <WiFiClient.h> //librari esp32 dijadikan client  
#include <BlynkSimpleEsp32.h> //library blink esp32  
#include <ESP32Servo.h>  
#include <Wire.h>  
#include <Adafruit_Sensor.h>  
#include <Adafruit_HMC5883_U.h>  
#include <SoftwareSerial.h>  
  
Servo servoKa;  
Servo servoKi;  
  
#define ka_maju 33  
#define ka_mundur 25  
#define ki_maju 27  
#define ki_mundur 26  
#define buzzer 19  
  
char auth[] = "FhCacjkZsHPGPqrbELRXhkJMV9WIAMav"; //kode  
autentikasi blynk  
char ssid[] = "Yolanda30"; //nama wifi  
char pass[] = "Navigasi123"; //password wifi  
int up, down, left, right, gate;  
  
int loadvoltage = 0;
```

```

float persenBaterai;
int speedKmph;
int pwm;
int servoPinka = 13;
int servoPinki = 32;
int deg;
int bateraiMonitor;
int setXYawal;

BlynkTimer timer; //deklarasi timer pada blynk
Adafruit_HMC5883_Unified mag =
Adafruit_HMC5883_Unified(12345);
void myTimerEvent() {
  // Blynk.virtualWrite(V5, persenBaterai); //kirim data persen baterai
  ke blynk
  Blynk.virtualWrite(V6, persenBaterai); //kirim data tegangan
  Blynk.virtualWrite(V7, bateraiMonitor); //kirim data status batt
  Blynk.virtualWrite(V9, longitude);
  Blynk.virtualWrite(V10, latitude);
  Blynk.virtualWrite(V11, deg);
}
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  if (!mag.begin()) {
    /* There was a problem detecting the HMC5883 ... check your
connections */
    Serial.println("Oops, no HMC5883 detected ... Check your
wiring!");
    while (1)
      ;
  }
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80); //inisialisasi blynk
cloud pada port 80
  timer.setInterval(1000L, myTimerEvent); //setting timer dengan
interfal 1detik atau 1000ms
  pinMode(ka_maju, OUTPUT);
  pinMode(ka_mundur, OUTPUT);
  pinMode(ki_maju, OUTPUT);

```

```

pinMode(ki_mundur, OUTPUT);
pinMode(34, INPUT); //volmeter
pinMode(buzzer, OUTPUT);
digitalWrite(buzzer, LOW);

ESP32PWM::allocateTimer(0);
ESP32PWM::allocateTimer(1);
ESP32PWM::allocateTimer(2);
ESP32PWM::allocateTimer(3);
servoKa.setPeriodHertz(50); // Standard 50hz servo
servoKi.setPeriodHertz(50); // Standard 50hz servo
servoKa.attach(servoPinka, 1000, 2000);
servoKi.attach(servoPinki, 1000, 2000);
servoKa.write(0);
// servoKi.write(90);

// using SG90 servo min/max of 500us and 2400us
// for MG995 large servo, use 1000us and 2000us,
// which are the defaults, so this line could be
// "myservo.attach(servoPin);"
}

void loop() {
  Blynk.run();
  timer.run();
  logicManual();
  readingAll();
}

void readingAll() {
  // current_mA = ina219.getCurrent_mA();
  // power_mW = ina219.getPower_mW();
  loadvoltage = analogRead(34);
  persenBaterai = map(loadvoltage, 0, 2670, 0, 12);
  bateraiMonitor = map(persenBaterai, 7, 12, 0, 100);
  sensors_event_t event;
  mag.getEvent(&event);
  float heading = atan2(event.magnetic.y, event.magnetic.x);
  float declinationAngle = 0.22;

```

```

heading += declinationAngle;
// Correct for when signs are reversed.
if (heading < 0) {
    heading += 2 * PI;
}
// Check for wrap due to addition of declination.
if (heading > 2 * PI) {
    heading -= 2 * PI;
}
int headingDegrees = heading * 180 / M_PI;
deg = headingDegrees;
if (deg > 180) {
    deg -= 360;
}
unsigned long pre = 0;
void printing() {
    if ((millis() - pre) >= 1000) {
        // Serial.print("up=" + String(up));
        // Serial.print(" down=" + String(down));
        // Serial.print(" left=" + String(left));
        // Serial.print(" right=" + String(right));
        // Serial.println(" gate=" + String(gate));
        //Serial.println("voltAdc=" + String(loadvoltage));
        // Serial.println("long=" + String(longitude, 5));
        // Serial.println("lat=" + String(latitude, 5));
        // Serial.println("deg=" + String(deg));
        Serial.println("Voltbat=" + String(persenBaterai));
        // Serial.println("persen=" + String(bateraiMonitor));
        // Serial.println("la_awal=" + String(lat_awal, 5));
        // Serial.println("lo_awal=" + String(lon_awal, 5));
        // Serial.println("degAwal=" + String(arahAwal));
        // Serial.println(" ");
        pre = millis();
    }
}
void maju() {
    analogWrite(ka_maju, pwm);
    analogWrite(ka_mundur, 0);
    analogWrite(ki_maju, pwm);
}

```

```

    analogWrite(ki_mundur, 0);
}
void mundur() {
    analogWrite(ka_maju, 0);
    analogWrite(ka_mundur, pwm);
    analogWrite(ki_maju, 0);
    analogWrite(ki_mundur, pwm);
}
void kanan() {
    analogWrite(ka_maju, 0);
    analogWrite(ka_mundur, 0);
    analogWrite(ki_maju, pwm);
    analogWrite(ki_mundur, 0);
}
void kiri() {
    analogWrite(ka_maju, pwm);
    analogWrite(ka_mundur, 0);
    analogWrite(ki_maju, 0);qq
    analogWrite(ki_mundur, 0);
}
void stp() {
    analogWrite(ka_maju, 0);
    analogWrite(ka_mundur, 0);
    analogWrite(ki_maju, 0);
    analogWrite(ki_mundur, 0);
}
void putar() {
    analogWrite(ka_maju, pwm);
    analogWrite(ka_mundur, 0);
    analogWrite(ki_maju, 0);
    analogWrite(ki_mundur, pwm);
}
void logicManual() {
    if (up == 1) {
        maju();
    } else if (down == 1) {
        mundur();
    } else if (left == 1) {
        kiri();
    }
}

```

```

} else if (right == 1) {
    kanan();
} else {
    stp();
}
if (gate == 1) {
    servoKa.write(230);
    // servoKi.write(90);
} else {
    servoKa.write(0);
    // servoKi.write(0);
}
if (bateraiMonitor < 20) {
    digitalWrite(buzzer, HIGH);
} else {
    digitalWrite(buzzer, LOW);
}
}
BLYNK_WRITE(V0) {
    up = param.asInt();
}
BLYNK_WRITE(V1) {
    down = param.asInt();
}
BLYNK_WRITE(V2) {
    left = param.asInt();
}
BLYNK_WRITE(V3) {
    right = param.asInt();
}
BLYNK_WRITE(V4) {
    gate = param.asInt();
}
BLYNK_WRITE(V8) {
    pwm = param.asInt();
}
BLYNK_WRITE(V5) {
    setXYawal = param.asInt();
}

```

**LAMPIRAN B**  
**DOKUMENTASI ALAT**



## BIODATA PENULIS



Nama : Yolanda Permata Sari  
Tempat/Tanggal Lahir : Cilacap, 28 Februari 2002  
Agama : Islam  
Alamat : Jalan Bali gg ontoseno, RT06/RW05  
Kecamatan Cilacap Utara, Kab. Cilacap  
Email : yolanda456@gmail.com  
Hobi : Berenang dan Traveling  
Motto : Sebaik-baiknya orang adalah bermanfaat  
bagi orang lain

### Riwayat Pendidikan

Sekolah/Institusi/ Universitas	Jurusan	Periode
SD Patra Mandiri	-	2008-2014
SMP N 2 Cilacap	-	2014-2017
SMA N 2 Cilacap	Ilmu Pengetahuan Sosial	2017-2020
Politeknik Negeri Cilacap	D3 Teknik Elektronika	2021-2024

Penulis telah mengikuti Sidang Tugas Akhir pada 9 Agustus 2024 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md).