DAFTAR PUSTAKA

Artiyasa, M., Idah N. R, Edwinanto, Anggy P.J. (2020). Aplikasi Smart Home NODE MCU IOT untuk Blynk. *Jurnal Rekayasa Teknologi Nusa Putra*. 7(1). 1-7. file:///C:/Users/Asus/Downloads/59-Article%20Text-143-2-10-20210302.pdf

Astuti, P., & Masdi, H. (2022). Sistem Kendali Kecepatan Motor BLDC Menggunakan PWM Berbasis Mikrokontroler Arduino Uno. JTEIN: *Jurnal Teknik Elektro Indonesia*, 3(1), 120-135. https://doi.org/10.24036/jtein.v3i1.216

Baisrum., Adnan R.A.T., Sarjono W.J. (2019). Kendali Kecepatan Motor DC Berbasis WNCS Menggunakan Pengendali PI Anti-Windup. *JTERA (Jurnal Teknologi Rekayasa).* 4(2). 227-236. http://dx.doi.org/10.31544/jtera.v4.i2.2019.227-236

Bintoro, Andik. (2022). Sistem Monitoring Air Pada Tangki Berbasis Internet of Things (IoT) Blynk *App. Universitas Malikussaleh*. https://www.researchgate.net/publication/366987643

Candra, T.Y., dan Ta'ali. (2020). Sistem Pengendali Kecepatan Motor DC Penguatan Terpisah Berbeban dengan Teknik Kontrol PWM Berbasis Arduino. *Jurnal Teknik Elektro dan Vokasional*. 6(1). 2302-3309. https://doi.org/10.24036/jtev.v6i1.107877

Fajri, Rizkia Meilani. (2022). Prototype Sistem Kontrol dan Monitoring Pergerakan Wiper, Starter, dan Kaca Jendela Pada Mobil Listrik Menggunakan Perintah Suara dengan Sensor Bluetooth dan Wifi Berbasis IOT. *Politeknik Manufaktur Negeri Bangka Belitung.* http://repository.polman-babel.ac.id/id/eprint/549/1/MAKALAH%20PA%20(FAJRI%20DAN%20RIZKIA%20MEILANI).pdf

JogjaRobotika, diakses pada 18 Agustus 2024, pukul 17.10, https://jogjarobotika.com/motor-DC-waterair-pumpaksesori/620-motor-DC-gearbox-kuning-148-3-72v.html

Kurniawan, F.F. Prabakti Endramawan, dan Denny Hardiyanto. (2022). Rancang Bangun Pengatur Kecepatan Motor DC Dengan PWM Berbasis Arduino Nano. *Jurnal Pendidikan Teknik Elektro*. 7(2). 9-16. http://doi.org/10.25273/jupiter.v7i2.14028

Kurniawan, Hendry. (2020). Pulse Width Modulation (PWM). Diakses pada 21 September 2024, pukul 23.12. https://www.elektronikahendry.com/2020/10/pulse-width-modulation-pwm.html

Ma'arif, Alfian. Ryan Istiarno., dan Sunardi. (2021). Kontrol Proporsional Integral Derivatif (PID) pada Kecepatan Sudut Motor DC dengan Pemodelan Identifikasi Sistem dan Tuning. *Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*. 9(2). 374-388. 4303-11052-1-PB.pdf

Nizam, M., Yuana, H., & Wulansari, Z. (2022). Mikrokontroler ESP 32 sebagai Alat Monitoring Pintu Berbasis Web. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2), 767-775. file:///C:/Users/Asus/Downloads/5713-Article%20Text-16517-2-10-20221101%20(1).pdf

Nura, Misbah Sulaiman., Yuwaldi Away., dan Fardian. (2020). Desain Sistem Kendali dan Monitoring Berbasis Teknologi WiFi untuk Otomasi Ruang Kerja Dosen. KITEKTRO : *Jurnal Online Teknik Elektro*. 5(3). 1-9. https://doi.org/10.24815/kitektro.v5i3.17254

Nurdin, M., Daud. (2016). Sistem Komunikasi Nirkabel. Bahan Ajar Kuliah. *Jurusan Teknik Elektro*. TEE 843. Universitas Mallikussaleh. https://repository.unimal.ac.id/2596/11/ST_09_Komunikasi-Nirkabel.pdf

Pangaribuan, Kosmas. (2019). Sistem Pengendali Motor DC Aplikasi Lift Dengan Pengendali Digital Berbasis Arduino. *Skripsi*. Universitas HKBP Nommensen. http://repository.uhn.ac.id/handle/123456789/3378

Pradana, Restu Adi. (2019). Perancangan Trainer Interface Mikrokontroler Berbasis ESP32 Sebagai Media Pembelajaran Pada Mata Kuliah

Interfacing. *Skripsi*. Universitas Raharja. https://widuri.raharja.info/index.php?title=SI1733499446

Prihatmoko, C. R. (2021). "Pengembangan Teknologi Smart Hybrid Reader Untuk Sistem Smart Campus Unhas". *Thesis (Skripsi)*. Universitas Hasanuddin. http://repository.unhas.ac.id:443/id/eprint/5589

S., Rifdian. I., & Hartono, H. (2018). Rancang Bangun Pulse Width Modulation (PWM) Sebagai Pengatur Kecepatan Motor DC Berbasis Mikrokontroler Arduino. *Jurnal Penelitian*. 3(1). 50-58. https://doi.org/10.46491/jp.v3i1.31

Septyawan, Tri. (2022). Rancang Bangun Keamanan Ruangan Pribadi dengan Arduino dan SMS Gateway. 2(3). 3. http://portaldata.org/index.php/portaldata/issue/view/6

Setiawan, David. (2017). Sistem Kontrol Motor DC Menggunakan PWM Arduino Berbasis Android System. *Jurnal Sains, Teknologi dan Industri*. 15(1). 7-14. http://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/2337

Suhendra, Tonny., Alena Uperiati., Dwi Amalia Purnamasari., dkk. (2018). Kendali Kecepatan Motor dengan Metode Pulse Width Modulation menggunakan N-channel Mosfet. Jurnal Sustainable : Jurnal Hasil Penelitian dan Industri Terapan. 7(2). 78-85. https://media.neliti.com/media/publications/266846-kendali-kecepatan-motor-DC-dengan-metode-a7110155.pdf

Wibowo, B.C., Aris Triwiyatno dan Sudjadi. (2023). Perancangan Pengaturan Kecepatan Motor DC Pada Mesin Sablon Kaos dengan Pulse Width Modulation (PWM). *Transient : Jurnal Ilmiah Teknik Elektro*. 12(1). 39-47. https://doi.org/10.14710/transient.v12i1.39-47

Wicaksono, Mochamad Hanif Dwi., dan Widi Aribowo. (2019). Pengendalian Motor DC Menggunakan Arduino Uno Pada Rancang Bangun Electrostatic Precipitator. *Indonesian Journal of Electrical and Electronics Engineering*. 2(2). 63-67. https://doi.org/10.26740/inajeee.v2n2.p29-33

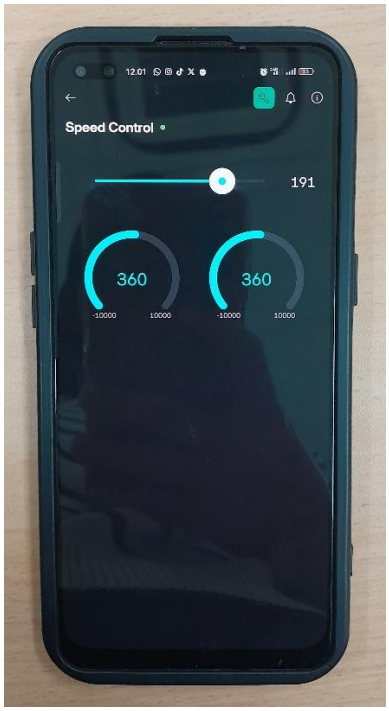
Winarno, Yoyok., dan Endryansyah. (2020). Sistem Kontrol Kecepatan Motor DC Pada Lift Konvensional Dengan Kontrol Linier Quadratic Regulator (LQR) Berbasis Arduino Uno. *Jurnal Teknik Elektro*. 9(2). 375-383. Garuda - Garba Rujukan Digital (kemdikbud.go.id)

Yericsen, Petra., Faisal Mahmuddin., dan Syerly Klara. (2023). Analisa Efisiensi Gearbox pada Motor Penggerak Listrik Kapal Nelayan. Jurnal Riset & Teknologi Terapan Kemaritiman. 2(1). 26-32. file:///C:/Users/Asus/Downloads/27933-Article%20Text-97918-2-10-20231001%20(4).pdf

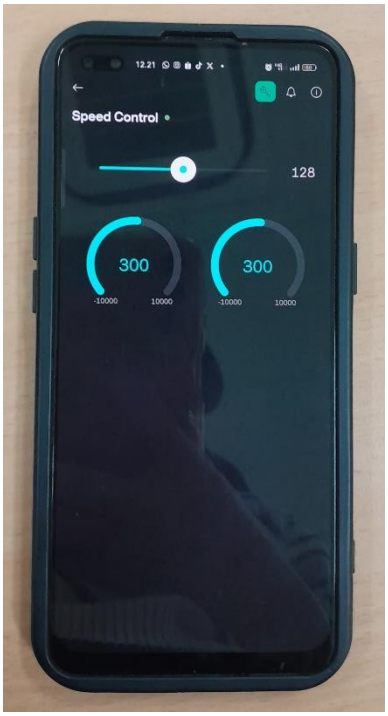
LAMPIRAN

Lampiran 1 Hasil Pengukuran Kecepatan Motor DC Mneggunakan Blynk *App*

Duty Cycle 100%



Duty cycle 75%



Duty Cycle 50%

Duty cycle 25%

Duty Cycle 0%

Lampiran 2 Hasil Pengukuran Kecepatan Motor DC Menggunakan Tachometer

Duty Cyle 100%

Motor 1

Motor 2

Duty Cycle 75%

Motor 1

Motor 2

Duty Cyle 50%

Motor 1

Motor 2

Duty Cyle 25%

Motor 1 Motor 2

Duty Cyle 0%

Motor 1 Motor 2

Lampiran 3 Pemrograman Master ESP32

```
/*************************************************************
  Blynk is a platform with iOS and Android Apps to control
  ESP32, Arduino, Raspberry Pi and the likes over the Internet.
  You can easily build mobile and web interfaces for any
  projects by simply dragging and dropping widgets.

    Downloads, docs, tutorials: https://www.blynk.io
    Sketch generator:           https://examples.blynk.cc
    Blynk community:            https://community.blynk.cc
    Follow us:                  https://www.fb.com/blynkApp
                                https://twitter.com/blynk_App

  Blynk library is licensed under MIT license
  This example code is in public domain.

 *************************************************************
  This example runs directly on ESP32 chip.

  NOTE: This requires ESP32 support package:
    https://github.com/espressif/arduino-esp32

  Please be sure to select the right ESP32 module
  in the Tools -> Board menu!

  Change WiFi ssid, pass, and Blynk auth token to run :)
  Feel free to Apply it to any other example. It's simple!
 *************************************************************/

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

/* Fill in information from Blynk Device Info here */
//#define BLYNK_TEMPLATE_ID           "TMPxxxxxx"
//#define BLYNK_TEMPLATE_NAME         "Device"
//#define BLYNK_AUTH_TOKEN            "YourAuthToken"
#define BLYNK_AUTH_TOKEN "NQEXMmLlVLhufIorc3oiba2nBlg4i5In"
#define BLYNK_TEMPLATE_ID "TMPL6kzVsshfp"
#define BLYNK_TEMPLATE_NAME "SPEED CONTROL"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
// Your WiFi credentials.
```

```
// Set password to "" for open networks.
char ssid[] = "bakpia";
char pass[] = "temanduduk";

float rpm = 0;
float rpm2 = 0;
//int pulseCount = 0;
unsigned long lastWifiCheck = 0;
const unsigned long wifiCheckInterval = 5000; // Interval untuk
mengecek koneksi WiFi (5 detik)

void setup()
{
  // Debug console
  Serial.begin(115200);
  //Serial.begin(9600);

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
}

void loop()
{
  Blynk.run();
  // Periksa koneksi WiFi setiap interval tertentu
  if (millis() - lastWifiCheck >= wifiCheckInterval) {
    if (WiFi.status() != WL_CONNECTED) {
      Serial.println("STOP"); // Kirim sinyal untuk mematikan motor
    }
    lastWifiCheck = millis();
  }
  if (Serial.available() > 0) {
    String data = Serial.readStringUntil('\n');
    if (data.startsWith("RPM1:")) {
      rpm = data.substring(5).toFloat();
    //rpm = Serial.parseFloat();
      Blynk.virtualWrite(V3, rpm); // Mengirimkan nilai RPM ke
aplikasi Blynk
    }
    else if (data.startsWith("RPM2:")){
      rpm2 = data.substring(5).toFloat();
      Blynk.virtualWrite(V4, rpm2);
    }
  }
```

```
  /*if (Serial.available() > 0) {
    rpm2 = Serial.parseFloat();
    Blynk.virtualWrite(V4, rpm2); // Mengirimkan nilai RPM ke
aplikasi Blynk
    //Serial.println(rpm);
  }*/
  //Serial.println(rpm);
}
BLYNK_WRITE(V2) {
  int SPEED = param.asInt();
  //Serial.print("MOTOR:");
  Serial.println(SPEED); // Kirim data ke Arduino Uno
}
//Blynk.virtualWrite(V3, rpm); // Mengirimkan nilai RPM ke aplikasi
Blynk
```

Lampiran 4 Pemrograman Slave Arduino

```
int motorPin1 = 8; // Pin PWM untuk kontrol kecepatan motor
int motorPin2 = 9; // Pin untuk arah motor
int motorPWM1 = 10;
int motorPin3 = 7; // Pin PWM untuk kontrol kecepatan motor
int motorPin4 = 6; // Pin untuk arah motor
int motorPWM2 = 5;
int sensorPin = 2;
int sensorPin2 = 3;
/*volatile int pulseCount = 0;
unsigned long lastTime = 0;*/
float rpm = 0;
float rpm2 = 0;
unsigned long millisBefore;
unsigned long millisBefore2;
volatile int pulseCount;
volatile int pulseCount2;
//int SPEED = 0;
void setup() {
  Serial.begin(115200); // Koneksi serial ke ESP32
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(motorPWM1, OUTPUT);
  pinMode(motorPin3, OUTPUT);
  pinMode(motorPin4, OUTPUT);
  pinMode(motorPWM2, OUTPUT);
  pinMode(sensorPin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(sensorPin), countPulse,
FALLING);
  pinMode(sensorPin2, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(sensorPin2), countPulse2,
FALLING);


}

void loop() {
  //int SPEED = Serial.parseInt();
  if (Serial.available() > 0) {
    String input = Serial.readStringUntil('\n');
    if (input == "STOP") {
      digitalWrite(motorPin1, HIGH);
      digitalWrite(motorPin2, LOW);
      analogWrite(motorPWM1, 0);
```

```
      digitalWrite(motorPin3, HIGH);
      digitalWrite(motorPin4, LOW);
      analogWrite(motorPWM2, 0);
    }
    else{
      int SPEED = input.toInt();
      digitalWrite(motorPin1, HIGH);
      digitalWrite(motorPin2, LOW);
      analogWrite(motorPWM1, SPEED);
      digitalWrite(motorPin3, HIGH);
      digitalWrite(motorPin4, LOW);
      analogWrite(motorPWM2, SPEED);
    }
    /*digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    analogWrite(motorPWM1, SPEED);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin4, LOW);
    analogWrite(motorPWM2, SPEED);*/
    //digitalWrite(motorPin2, SPEED > 0 ? HIGH : LOW);
  }
  /*calculateRPM();
  if (millis() - lastTime >= 1000) { // Update RPM setiap 1 detik
    //Serial.print("RPM: ");
    //Serial.println(rpm);
    //Serial.print("R");
    Serial.println(rpm); // Mengirimkan RPM ke ESP32 dengan format
khusus
    lastTime = millis();
  }*/
  if (millis() - millisBefore >= 1000){
    rpm = (pulseCount / 20)*60;
    pulseCount = 0;
    millisBefore = millis();
    Serial.print("RPM1:");
    Serial.println(rpm); //Serial.println(rpm2);
  }
  if (millis() - millisBefore2 >= 1000){
    rpm2 = (pulseCount2 / 20)*60;
    pulseCount2 = 0;
    millisBefore2 = millis();
    Serial.print("RPM2:");
    Serial.println(rpm2);
```

```
  }
  //delay(100);
}

void countPulse() {
  pulseCount++;
}
void countPulse2() {
  pulseCount2++;
}

/*void calculateRPM() {
  unsigned long currentTime = millis();
  unsigned long elapsedTime = currentTime - lastTime;
  if (elapsedTime >= 1000) { // Hitung RPM setiap 1 detik
    rpm = (pulseCount * 60.0) / (elapsedTime / 1000.0);
    pulseCount = 0;
    lastTime = currentTime;
  }
}*/
```