

DAFTAR PUSTAKA

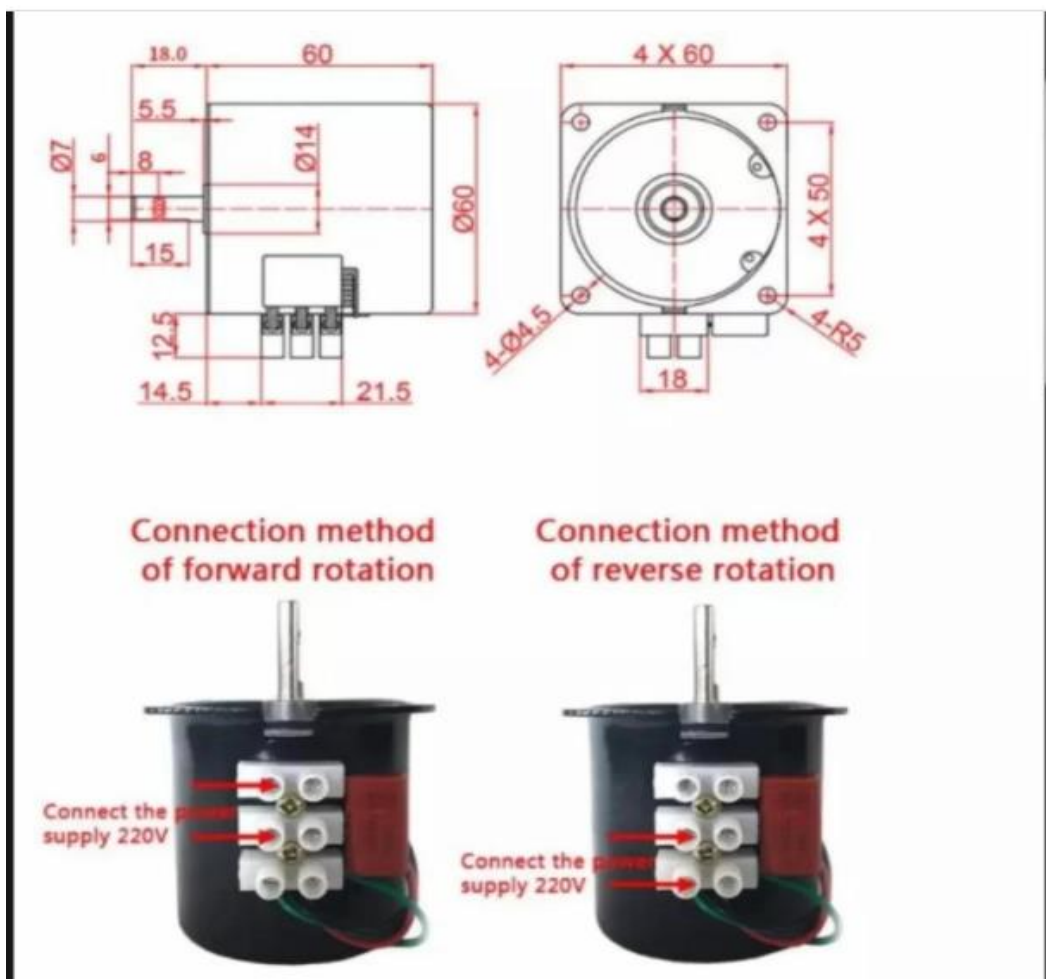
- Arifin AR, B.E. 2010. Fisiologi Tidur dan Pernafasan. *Jurnal Respiologi*: 1–12.
- Barone, T.L. 2017. “Sleep is on the back burner”: Working students and sleep. *Social Science Journal*, 54(2).
- Chen, W.L. dan Chen, J.H. 2019. Consequences of inadequate sleep during the college years: Sleep deprivation, grade point average, and college graduation. *Preventive Medicine*, 124.
- Cui, L., Xue, R., Zhang, X., Chen, S., Wan, Y. dan Wu, W. 2019. Sleep deprivation inhibits proliferation of adult hippocampal neural progenitor cells by a mechanism involving IL-17 and p38 MAPK. *Brain Research*, 1714.
- Dhopir, Muh Ilham Ali and Prasetyo, M.D. 2016. Rancang bangun alat otomatisasi pembuatan beton berbasis PLC. *Journal of Chemical Information and Modeling*.
- Diarsyah Amarullah, Mochammad Djaohar dan Massus Subekti. 2020. PENGATURAN KECEPATAN MOTOR DC SERI BERBASIS ARDUINO UNO. *Journal of Electrical Vocational Education and Technology*, 4(2).
- Egi, E., Prastiwi, S. dan Putri, R.M. 2017. Hubungan Gangguan Tidur dengan Tingkat Kesegaran Jasmani Remaja Putri di SMK Kertha Wisata Kelurahan Tlogomas Kecamatan Lowokwaru Malang. *Nursing News*, 2(1).
- Foss, R.D., Smith, R.L. dan O’Brien, N.P. 2019. School start times and teenage driver motor vehicle crashes. *Accident Analysis and Prevention*, 126.
- Frenda, S.J. dan Fenn, K.M. 2016. Sleep Less, Think Worse: The Effect of Sleep Deprivation on Working Memory. *Journal of Applied Research in Memory and Cognition*, 5(4).
- Gangwisch, J.E., Heymsfield, S.B., Boden-Albala, B., Buijs, R.M., Kreier, F., Pickering, T.G., Rundle, A.G., Zammit, G.K. dan Malaspina, D. 2006. Short sleep duration as a risk factor for hypertension: Analyses of the first National Health and Nutrition Examination Survey. *Hypertension*, 47(5).
- Hughes, A. dan Drury, B. 2013. *Electric Motors and Drives*.

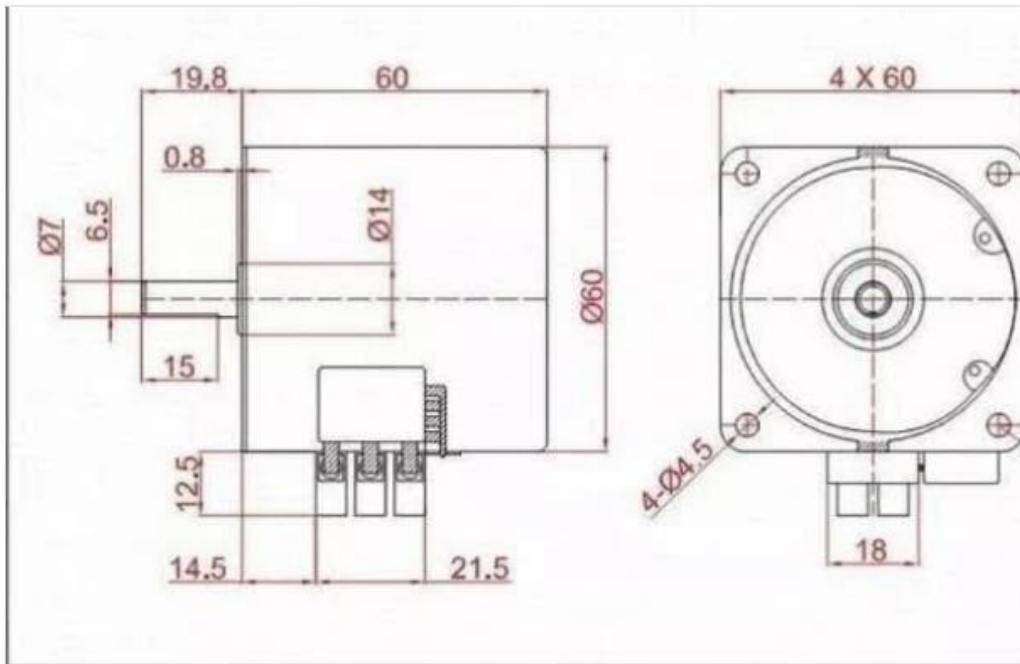
- Iven Kartadinata, S.T. 2008. I Love You Tomorrow: Prokrastinasi Akademik dan Manajemen Waktu. *Indonesian Psychological Journal*, 23(2).
- Jung, J.H., Kim, M., Lee, S.J., Lee, E., Lee, S.A., Lee, J.D., Choi, J.H. dan Kim, B.G. 2018. Effect of sleep deprivation on hearing levels in rats. *International Journal of Pediatric Otorhinolaryngology*, 112.
- Kadir, A. 2013. Pengertian Arduino. *Arduino*, (1).
- Lund, H.G., Reider, B.D., Whiting, A.B. dan Prichard, J.R. 2010. Sleep Patterns and Predictors of Disturbed Sleep in a Large Population of College Students. *Journal of Adolescent Health*, 46(2).
- Mah, C.D., Kezirian, E.J., Marcello, B.M. dan Dement, W.C. 2018. Poor sleep quality and insufficient sleep of a collegiate student-athlete population. *Sleep Health*, 4(3).
- Nault, W.H. 2006. No Title. In *The World Book Encyclopedia*, B - Volume 2. Chicago: World Book: 198.
- Oxford University Press. 2020. *Oxford Advanced Learner's Dictionary (Seventh Edition)*. Oxford University Press.
- Potter, P.A. dan Perry, A.G. 2005. *Buku ajar fundamental keperawatan: konsep, proses, dan praktik*. Jakarta: EGC.
- Setiawan, D., Yos Sudarso Km, J., Kunci, K. dan Uno, A. 2017. Sistem Kontrol Motor Dc Menggunakan Pwm Arduino Berbasis Android System. *Jurnal Sains, Teknologi dan Industri*, 15(1).
- Shakkottai, A., O'Brien, L.M., Nasr, S.Z. dan Chervin, R.D. 2018. Sleep disturbances and their impact in pediatric cystic fibrosis. *Sleep Medicine Reviews*, 42.
- Simanjuntak, V.V. 2018. Analisis Dc Motor Pada Aplikasi Parkir Vertikal Otomatis Menggunakan Rfid. : 12–13. <http://eprints.polsri.ac.id/id/eprint/4649>.
- Sulianti, A., Studi Psikologi Fakultas Psikologi UIN Sunan Gunung Djati, P., Suseno, A., Verina, A., Naufal Fadlurrahman Riyadhi, M. dan Psikologi UIN Sunan Gunung Djati Bandung, F. 2020. Prokrastinasi dan Pola Tidur Mahasiswa. *Penelitian Psikologi*, Vol. 11 No: 67–75. <http://doi.org/10.29080/jpp.v11i2.454>.

- Tempesta, D., Socci, V., Coppo, M., Dello Ioio, G., Nepa, V., De Gennaro, L. dan Ferrara, M. 2016. The effect of sleep deprivation on the encoding of contextual and non-contextual aspects of emotional memory. *Neurobiology of Learning and Memory*, 131.
- Verkooijen, S., de Vos, N., Bakker-Camu, B.J.W., Branje, S.J.T., Kahn, R.S., Ophoff, R.A., Plevier, C.M. dan Boks, M.P.M. 2018. Sleep Disturbances, Psychosocial Difficulties, and Health Risk Behavior in 16,781 Dutch Adolescents. *Academic Pediatrics*, 18(6).
- Yuan, R., Wang, J. dan Guo, L. li. 2016. The Effect of Sleep Deprivation on Coronary Heart Disease. *Chinese Medical Sciences Journal*, 31(4).
- Zuhal. 1991. *Dasar Teknik Tenaga Listrik dan Elektronika Daya*. Jakarta: Gramedia.

LAMPIRAN

Datasheet Motor yang Digunakan





Technical Parameters of Synchronous Motor

Voltage 电压 (V)	Input Power 输入功率 (W)	Input Current 输入电流 (mA)	Starting Voltage 启动电压 (V)	Temperature Rise 温升 (K)	Noise Level 噪音 (dB)	Rotation Direction 旋转方向	外
100-120	≤14	≤110	(100-120) ± 15%	≤60	≤45	CW/CCW	
220-240	≤14	≤55	(220-240) ± 15%	≤60	≤45	CW/CCW	

Torque and Speed of Synchronous Motor

Rated speed 转速 (rpm)	2.5/3	3.8/4.5	5/6	7.5/9	10/12	12/15	15/18	20/24	25/30	30/36	40/48	50/60	60/72
Normal torque 普通力矩 (kgf.cm)	45/38	32/27	26/21.5	20/17	15/12	13.5/11	10/8.3	7.5/6	6.5/5.3	5/4.2	4/3.3	3/2.5	2.5/2
Higher torque 较大力矩 (kgf.cm)	60/50	50/40	40/34	25/21	20/17	18/15	14/11.5	10/8.3	8.5/7.2	7.5/6	6/5	4/3.3	3.5/3

Program Arduino

```
#include <LiquidCrystal.h>
```

```
#include <DS3231.h>
```

```
#include <Wire.h>
```

```
#define RS 8
```

```
#define EN 9
```

```
#define D4 4
```

```
#define D5 5
```

```
#define D6 6
```

```
#define D7 7
```

```
#define MENU (0)
```

```
#define UP (1)
```

```
#define DOWN (2)
```

```
#define OK (3)
```

```
#define BACK (4)
```

```
#define STATIC (5)
```

```
#define INTERVAL (1000)
```

```
#define BTN_INTERVAL (250)
```

```
DS3231 clk;
```

```
LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);
```

```
const int BTN_UP = 129;
```

```
const int BTN_DOWN = 305;
```

```
const int BTN_BACK = 478;
```

```
const int BTN_OK = 0;
```

```
const int BTN_MENU = 720;
```

```
bool century = false;
bool h12Flag;
bool pmFlag;

int hour = 0;
int minute = 0;

bool btn_pressed = false;
bool mode = false;
bool alarm = false;
bool alarm_mode = false;

byte selected_button;
byte last_selected_button = STATIC;

unsigned long counting;
unsigned long last_counting;
unsigned long btn_pressed_time;
unsigned long alarm_time;

byte clock_icon[] = {
    B01110,
    B10001,
    B10101,
    B10101,
    B10111,
    B10001,
    B01110,
    B00000
};
```

```

void set_selected_button(int val) {
    if (val >= BTN_UP - 5 && val <= BTN_UP + 5)
        selected_button = UP;
    else if (val >= BTN_DOWN - 5 && val <= BTN_DOWN + 5)
        selected_button = DOWN;
    else if (val >= BTN_MENU - 5 && val <= BTN_MENU + 5)
        selected_button = MENU;
    else if (val >= BTN_OK - 5 && val <= BTN_OK + 5)
        selected_button = OK;
    else if (val >= BTN_BACK - 5 && val <= BTN_BACK + 5)
        selected_button = BACK;
    else
        selected_button = STATIC;
}

void set_mode(bool mode) {
    if (mode == false) {
        lcd.setCursor(0, 0);
        lcd.print("Set Hour  : ");
        if (String(hour).length() < 2) {
            lcd.print(0);
            lcd.print(hour);
        } else {
            lcd.print(hour);
        }
    } else {
        lcd.setCursor(0, 0);
        lcd.print("Set Minute : ");
        if (String(minute).length() < 2) {
            lcd.print(0);
            lcd.print(minute);
        } else {

```



```

        lcd.print(minute);
    }
}
}

void print_time() {
    lcd.setCursor(0, 0);
    lcd.write(byte(0));
    lcd.print(" ");
    lcd.print(String(clk.getHour(h12Flag, pmFlag)).length() < 2 ? "0" +
String(clk.getHour(h12Flag, pmFlag)) : String(clk.getHour(h12Flag, pmFlag)));
    lcd.print(":");
    lcd.print(String(clk.getMinute()).length() < 2 ? "0" + String(clk.getMinute()) :
String(clk.getMinute()));
    lcd.print(" ");
    lcd.print(String(clk.getDate()).length() < 2 ? "0" + String(clk.getDate()) :
String(clk.getDate()));
    lcd.print("/");
    lcd.print(String(clk.getMonth(century)).length() < 2 ? "0" +
String(clk.getMonth(century)) : String(clk.getMonth(century)));
    lcd.print("/");
    lcd.print(String(clk.getYear()).length() < 2 ? "0" + String(clk.getYear()) :
String(clk.getYear()));
}

void print_alarm(bool state) {
    if (!state) {
        lcd.setCursor(0, 1);
        lcd.print("ALARM : (No Set)");
    } else {
        lcd.setCursor(0, 1);
        lcd.print("ALARM : ");
    }
}

```

```

if (String(hour).length() < 2) {
    lcd.print(0);
    lcd.print(hour);
} else {
    lcd.print(hour);
}
lcd.print(":");
if (String(minute).length() < 2) {
    lcd.print(0);
    lcd.print(minute);
} else {
    lcd.print(minute);
}
}
}

```

```

void inc_time(bool mode) {
    if (mode == true) {
        lcd.setCursor(13, 0);
        if (hour > 22) hour = -1;
        ++hour;
        if (String(hour).length() < 2) {
            lcd.print(0);
            lcd.print(hour);
            lcd.print(" ");
        } else {
            lcd.print(hour);
            lcd.print(" ");
        }
    } else {
        lcd.setCursor(13, 0);
    }
}

```

```

if (minute > 58) minute = -1;
++minute;
if (String(minute).length() < 2) {
    lcd.print(0);
    lcd.print(minute);
    lcd.print(" ");
} else {
    lcd.print(minute);
    lcd.print(" ");
}
}
}

```

```

void dec_time(bool mode) {
    if (mode == true) {
        lcd.setCursor(13, 0);
        if (hour < 1) hour = 24;
        --hour;
        if (String(hour).length() < 2) {
            lcd.print(0);
            lcd.print(hour);
            lcd.print(" ");
        } else {
            lcd.print(hour);
            lcd.print(" ");
        }
    }
} else {
    lcd.setCursor(13, 0);
    if (minute < 1) minute = 60;
    --minute;
}
}

```

```

if (String(minute).length() < 2) {
  lcd.print(0);
  lcd.print(minute);
  lcd.print(" ");
} else {
  lcd.print(minute);
  lcd.print(" ");
}
}
}

```

```

void setup() {
  Serial.begin(9600);
  Wire.begin();
  lcd.begin(16, 2);
  lcd.createChar(0, clock_icon);
  selected_button = STATIC;
  last_counting = 0;
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  digitalWrite(13, LOW);
  digitalWrite(12, HIGH);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Ranjang Tidur");
  lcd.setCursor(0, 1);
  lcd.print("Alarm #byFahri");
  delay(1500);
  print_time();
  print_alarm(alarm);
}

```

```
}
```

```
void loop() {
```

```
  set_selected_button(analogRead(A0));
```

```
  counting = millis();
```

```
  if (btn_pressed == false) {
```

```
    if (counting - last_counting >= INTERVAL) {
```

```
      last_counting = counting;
```

```
      print_time();
```

```
      print_alarm(alarm);
```

```
    }
```

```
  }
```

```
  if (alarm == true) {
```

```
    if (hour == clk.getHour(h12Flag, pmFlag) && minute == clk.getMinute()) {
```

```
      alarm_time = 0;
```

```
      lcd.clear();
```

```
      while (1) {
```

```
        set_selected_button(analogRead(A0));
```

```
        if (selected_button == OK) {
```

```
          digitalWrite(12, HIGH);
```

```
          digitalWrite(13, LOW);
```

```
          lcd.clear();
```

```
          print_time();
```

```
          print_alarm(alarm);
```

```
          alarm = false;
```

```
          break;
```

```
        }
```

```
        if ((unsigned long) (millis() - alarm_time) >= 500) {
```

```
          alarm_time = millis();
```

```

    alarm_mode = !alarm_mode;
    digitalWrite(13, alarm_mode);
}
lcd.setCursor(0, 0);
lcd.print("WAKE UP!!!!");
digitalWrite(12, LOW);
}
}
}

```

```

if ((selected_button == MENU) && (last_selected_button != MENU) &&
btn_pressed == false) {
    lcd.clear();
    set_mode(false);
    mode = false;
    btn_pressed = true;
}
if (btn_pressed == true) {
    while (btn_pressed) {
        set_selected_button(analogRead(A0));
        if ((selected_button != STATIC) && (selected_button != last_selected_button)
&& ((unsigned long) (millis() - btn_pressed_time) >= BTN_INTERVAL)) {
            btn_pressed_time = millis();
            if (mode == false) {
                switch (selected_button) {
                    case OK:
                        set_mode(true);
                        mode = true;
                        break;
                    case UP:
                        inc_time(1);

```

```

        break;
    case DOWN:
        dec_time(1);
        break;
    case BACK:
        alarm = false;
        btn_pressed = false;
        break;
    }
} else {
    switch (selected_button) {
        case OK:
            lcd.clear();
            Serial.println("Keluar...");
            alarm = true;
            btn_pressed = false;
            break;
        case UP:
            inc_time(0);
            break;
        case DOWN:
            dec_time(0);
            break;
        case BACK:
            set_mode(false);
            mode = false;
            break;
    }
}
}

```

```
    last_selected_button = selected_button;  
  }  
}
```

```
last_selected_button = selected_button;  
}
```


Dokumentasi Perakitan

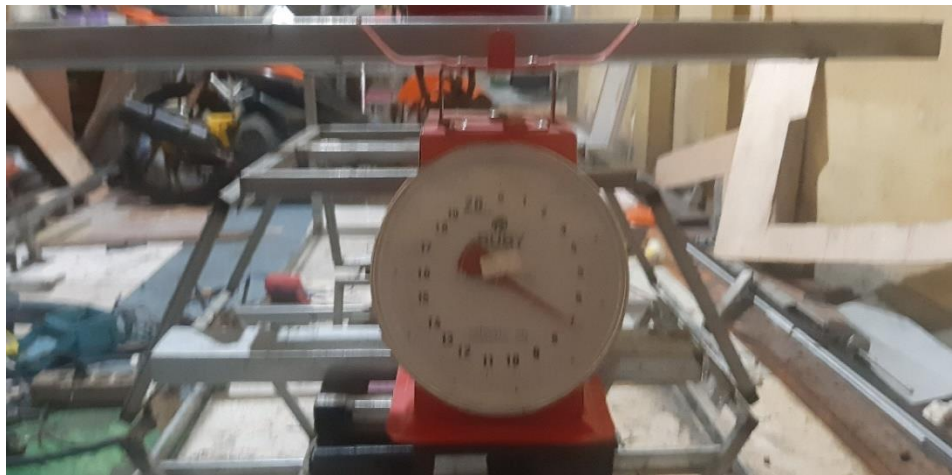


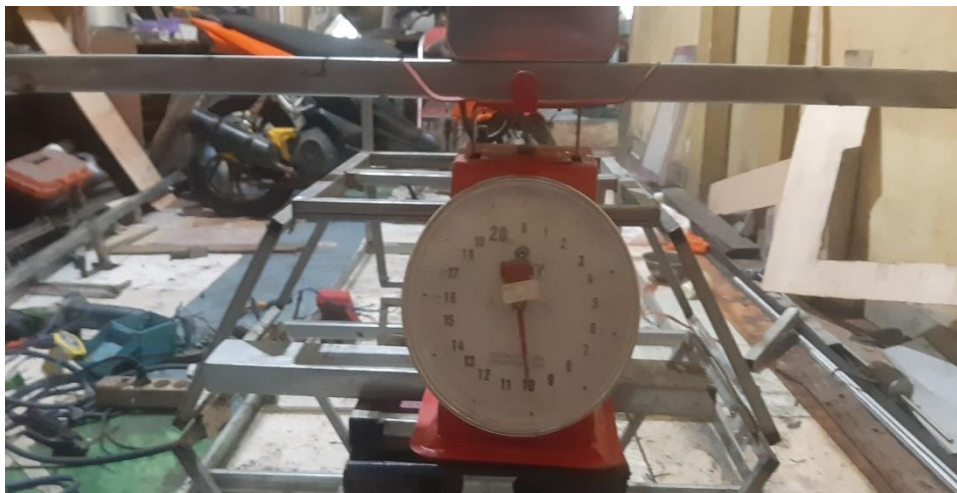
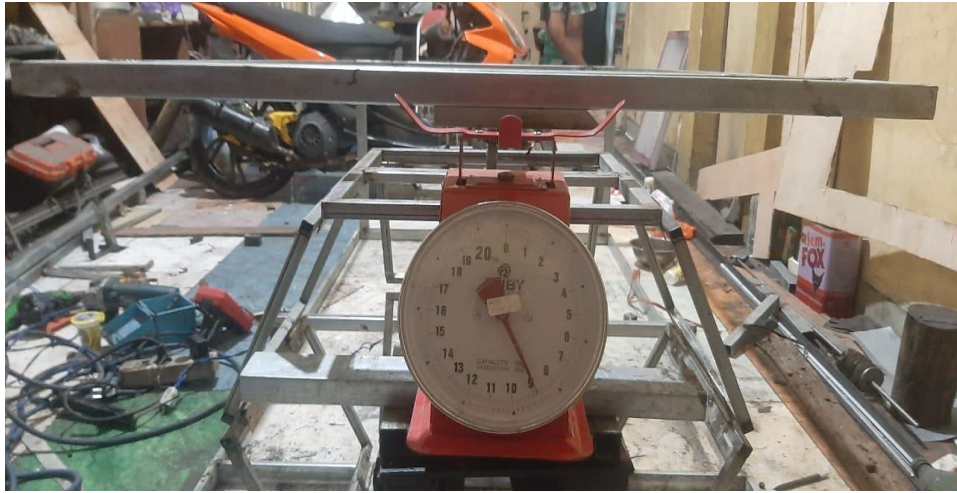
Dokumentasi Pengujian

1. Pengukuran Arus RMS Motor



2. Pengukuran Massa Beban pada Motor menggunakan Timbangan





3. Pengukuran Kecepatan Menggunakan *Tachometer*

