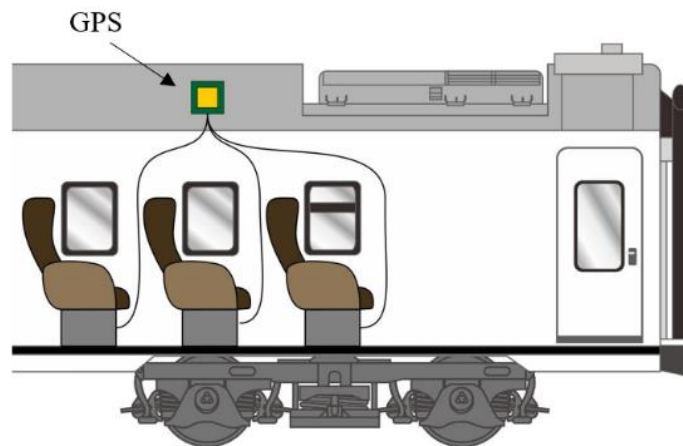
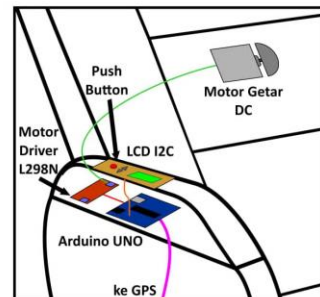
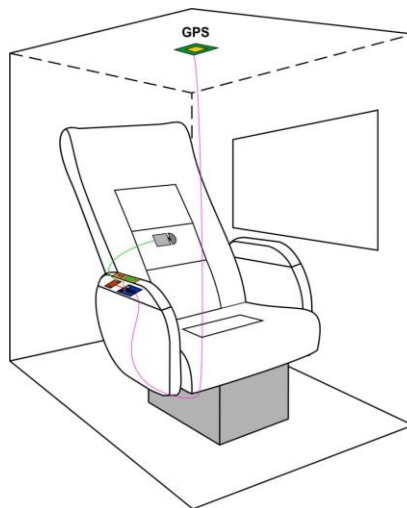
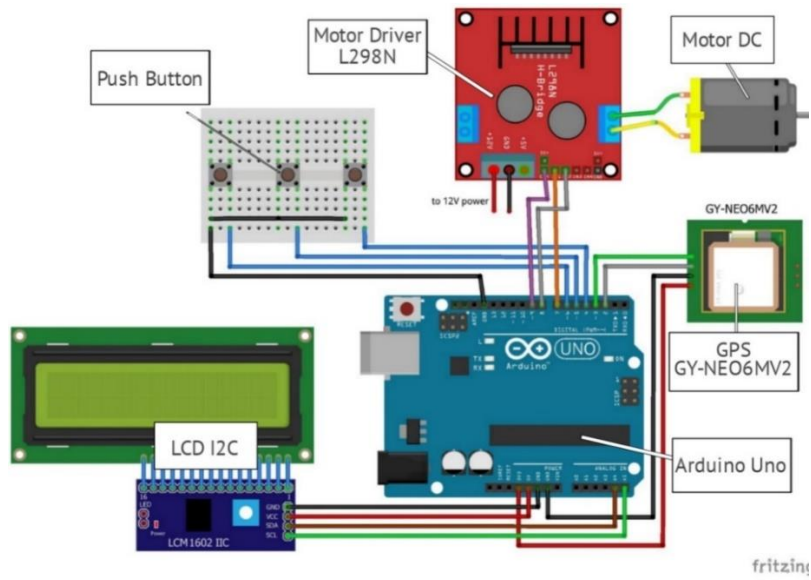


LAMPIRAN

Lampiran 1 Skematik dan Dsain Alat :



Lampiran 2 Coding :

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

//push button
const int tUP = 4;
const int tOK = 5;
const int tDN = 6;
int ok = 0;
int up = 0;
int down = 0;

LiquidCrystal_I2C lcd(0x27,16,2);

//gps
static const int RXPin = 3, TXPin = 2;
static const uint32_t GPSBaud = 9600;
TinyGPSPlus gps;
SoftwareSerial ss(RXPin, TXPin);

// For stats that happen every 5 seconds
unsigned long last = 0UL;

//pin motor driver
#define IN1 7
#define IN2 8
#define ENA 9

void setup()
{
  Serial.begin(9600);
  ss.begin(GPSBaud);

  pinMode(tUP, INPUT_PULLUP);
  pinMode(tOK, INPUT_PULLUP);
  pinMode(tDN, INPUT_PULLUP);

  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(1, 0);
  lcd.print("SELAMAT DATANG");
  lcd.setCursor(2, 1);
  lcd.print("TA Risman MM");
  delay(5000);
  lcd.clear();
}

void loop()
{
  Menu:
  while(1)

```

```

{
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("Tekan OK untuk");
  lcd.setCursor(2,1);
  lcd.print("Pilih Tujuan");
  delay(500);

  up = digitalRead(tUP);
  ok = digitalRead(tOK);
  down = digitalRead(tDN);

  if(ok == LOW)
  {
    delay(100);
    goto Menu1;
  }
}

// pilihan Perum
Menu1:
while(1)
{
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("PERUM GARUDA");
  lcd.setCursor(10,1);
  lcd.print("next >");
  delay(500);

  up = digitalRead(tUP);
  ok = digitalRead(tOK);
  down = digitalRead(tDN);
  if(ok == LOW)
  {
    delay(100);
    goto Menu3;
  }
  if(down == LOW)
  {
    delay(100);
    goto Menu2;
  }
}

// pilihan Gobras
Menu2:
while(1)
{
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("GOBRAS TASIK");
  lcd.setCursor(0,1);
  lcd.print("< prev");
  delay(500);

  up = digitalRead(tUP);
  ok = digitalRead(tOK);
  down = digitalRead(tDN);

```

```

    if(ok == LOW)
    {
        delay(100);
        goto Menu4;
    }

    if(up == LOW)
    {
        delay(100);
        goto Menu1;
    }
}

// pilihan getar untuk Perum
Menu3:
while(1)
{
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("Kekuatan Getar");
    lcd.setCursor(0,1);
    lcd.print("Min   Back   Max");
    delay(500);

    up = digitalRead(tUP);
    ok = digitalRead(tOK);
    down = digitalRead(tDN);

    if(down == LOW)
    {
        delay(100);
        lcd.clear();
        lcd.setCursor(2,0);
        lcd.print("SELAMAT JALAN");
        goto Menu6;
    }
    if(ok == LOW)
    {
        delay(100);;
        goto Menu1;
    }
    if(up == LOW)
    {
        delay(100);
        lcd.clear();
        lcd.setCursor(2,0);
        lcd.print("SELAMAT JALAN");
        goto Menu5;
    }
}

// pilihan getar untuk Goras
Menu4:
while(1)
{
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("Kekuatan Getar");
    lcd.setCursor(0,1);
    lcd.print("Min   Back   Max");

```

```

delay(500);

up = digitalRead(tUP);
ok = digitalRead(tOK);
down = digitalRead(tDN);

if(down == LOW)
{
    delay(100);
    lcd.clear();
    lcd.setCursor(2,0);
    lcd.print("SELAMAT JALAN");
    goto Menu8;
}
if(ok == LOW)
{
    delay(100);
    goto Menu2;
}
if(up == LOW)
{
    delay(100);
    lcd.clear();
    lcd.setCursor(2,0);
    lcd.print("SELAMAT JALAN");
    goto Menu7;
}
}
// Perum getar ringan
Menu5:
while(1)
{
    perum_min();

    up = digitalRead(tUP);
    ok = digitalRead(tOK);
    down = digitalRead(tDN);

    if(ok == LOW)
    {
        delay(100);
        goto Menu;
    }
}
// Perum getar kuat
Menu6:
while(1)
{
    perum_max();

    up = digitalRead(tUP);
    ok = digitalRead(tOK);
    down = digitalRead(tDN);

    if(ok == LOW)
    {
        delay(100);
        goto Menu;
    }
}

```

```

}
// Gobras getar ringan
Menu7:
while(1)
{
  gobras_min();

  up = digitalRead(tUP);
  ok = digitalRead(tOK);
  down = digitalRead(tDN);

  if(ok == LOW)
  {
    delay(100);
    goto Menu;
  }
}
// Gobras getar kuat
Menu8:
while(1)
{
  gobras_max();

  up = digitalRead(tUP);
  ok = digitalRead(tOK);
  down = digitalRead(tDN);

  if(ok == LOW)
  {
    delay(100);
    lcd.clear();
    goto Menu;
  }
}
}

void gobras_max()
{
  while (ss.available() > 0)
    gps.encode(ss.read());

  if (gps.location.isUpdated())
  {
    Serial.print(F(" Lat="));
    Serial.print(gps.location.lat(), 6);
    Serial.print(F(" Long="));
    Serial.println(gps.location.lng(), 6);
  }

  else if (millis() - last > 1000)
  {
    Serial.println();
    if (gps.location.isValid())
    {
      static const double gobras_LAT = -7.373085, gobras_LON =
108.232574;
      double distanceToDest =

```

```

    TinyGPSPlus::distanceBetween(
        gps.location.lat(),
        gps.location.lng(),
        gobras_LAT,
        gobras_LON);

    Serial.print(F("Distance to Destination ="));
    Serial.print(distanceToDest/1000, 6);

    if(distanceToDest/1000 < 0.500000)
    {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Anda Telah Tiba");
        lcd.setCursor(0,1);
        lcd.print("di Gobras Tasik");

        digitalWrite(IN1, HIGH);
        digitalWrite(IN2, LOW);
        analogWrite(ENA, 255);
    }
}

if (gps.charsProcessed() < 10)
    Serial.println(F("WARNING: No GPS data. Check wiring.));

last = millis();
Serial.println();
}

up = digitalRead(tUP);
ok = digitalRead(tOK);
down = digitalRead(tDN);

if(ok == LOW)
{
    delay(100);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    analogWrite(ENA, LOW);
}
}

void gobras_min()
{
    while (ss.available() > 0)
        gps.encode(ss.read());

    if (gps.location.isUpdated())
    {
        Serial.print(F(" Lat="));
        Serial.print(gps.location.lat(), 6);
        Serial.print(F(" Long="));
        Serial.println(gps.location.lng(), 6);
    }

    else if (millis() - last > 1000)

```

```

{
  Serial.println();
  if (gps.location.isValid())
  {
    static const double gobras_LAT = -7.373085, gobras_LON =
108.232574;
    double distanceToDest =
      TinyGPSPlus::distanceBetween(
        gps.location.lat(),
        gps.location.lng(),
        gobras_LAT,
        gobras_LON);

    Serial.print(F("Distance to Destination ="));
    Serial.print(distanceToDest/1000, 6);

    if(distanceToDest/1000 < 0.500000)
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Anda Telah Tiba");
      lcd.setCursor(0,1);
      lcd.print("di Gobras Tasik");

      digitalWrite(IN1, HIGH);
      digitalWrite(IN2, LOW);
      analogWrite(ENA, 150);
    }
  }

  if (gps.charsProcessed() < 10)
    Serial.println(F("WARNING: No GPS data. Check wiring.));

  last = millis();
  Serial.println();
}

up = digitalRead(tUP);
ok = digitalRead(tOK);
down = digitalRead(tDN);

if(ok == LOW)
{
  delay(100);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  analogWrite(ENA, LOW);
}
}

void perum_max()
{
  while (ss.available() > 0)
    gps.encode(ss.read());

  if (gps.location.isUpdated())
  {
    Serial.print(F(" Lat="));

```



```

    Serial.print(gps.location.lat(), 6);
    Serial.print(F(" Long="));
    Serial.println(gps.location.lng(), 6);

}

else if (millis() - last > 1000)
{
    Serial.println();
    if (gps.location.isValid())
    {
        static const double Perum_LAT = -7.379742, Perum_LON =
108.221336;
        double distanceToDest =
            TinyGPSPlus::distanceBetween(
                gps.location.lat(),
                gps.location.lng(),
                Perum_LAT,
                Perum_LON);

        Serial.print(F("Distance to Destination = "));
        Serial.print(distanceToDest/1000, 6);

        if(distanceToDest/1000 < 0.500000)
        {
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("Anda Telah Tiba");
            lcd.setCursor(0,1);
            lcd.print("di Perum Garuda");

            digitalWrite(IN1, HIGH);
            digitalWrite(IN2, LOW);
            analogWrite(ENA, 255);
        }
    }

    if (gps.charsProcessed() < 10)
        Serial.println(F("WARNING: No GPS data. Check wiring.));

    last = millis();
    Serial.println();
}

up = digitalRead(tUP);
ok = digitalRead(tOK);
down = digitalRead(tDN);

if(ok == LOW)
{
    delay(100);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    analogWrite(ENA, LOW);
}

}

void perum_min()

```

```

{
while (ss.available() > 0)
  gps.encode(ss.read());

if (gps.location.isUpdated())
{
  Serial.print(F("  Lat="));
  Serial.print(gps.location.lat(), 6);
  Serial.print(F(" Long="));
  Serial.println(gps.location.lng(), 6);
}

else if (millis() - last > 1000)
{
  Serial.println();
  if (gps.location.isValid())
  {
    static const double Perum_LAT = -7.379742, Perum_LON =
108.221336;
    double distanceToDest =
      TinyGPSPlus::distanceBetween(
        gps.location.lat(),
        gps.location.lng(),
        Perum_LAT,
        Perum_LON);

    Serial.print(F("Distance to Destination = "));
    Serial.print(distanceToDest/1000, 6);

    if(distanceToDest/1000 < 0.500000)
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Anda Telah Tiba");
      lcd.setCursor(0,1);
      lcd.print("di Perum Garuda");

      digitalWrite(IN1, HIGH);
      digitalWrite(IN2, LOW);
      analogWrite(ENA, 150);
    }
  }

  if (gps.charsProcessed() < 10)
    Serial.println(F("WARNING: No GPS data. Check wiring.));

  last = millis();
  Serial.println();
}

up = digitalRead(tUP);
ok = digitalRead(tOK);
down = digitalRead(tDN);

if(ok == LOW)
{
  delay(100);
  digitalWrite(IN1, LOW);
}

```

```
digitalWrite(IN2, LOW);  
analogWrite(ENA, LOW);  
}  
}
```

Lampiran 3 Dokumentasi :

