



MANUAL BOOK ALAT CRASH DETECTION UNTUK MOBIL BERBASIS IOT

Eka Mahendra Bagaskara
2241727009

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

DAFTAR ISI

DAFTAR ISI	1
DAFTAR GAMBAR	ii
DAFTAR TABEL	iv
BAB I PENDAHULUAN.....	1
1.1 Gambaran Umum	1
1.2 Komponen	1
1.3 Persyaratan Sistem	3
BAB II PERSIAPAN HARDWARE.....	4
2.1 Koneksi ESP32 Ke Komputer.....	4
2.2 Koneksi Sensor MPU6050	4
2.3 Koneksi Sensor GPS U-Blox Neo6.....	4
2.4 Koneksi OLED LCD.....	4
2.5 Rangkaian Komponen ke dalam Mobil Remote Kontrol.....	4
BAB III INSTALASI PERANGKAT LUNAK	6
3.1 Instalasi Arduino IDE.....	6
3.2 Instal Library Pada Arduino IDE	11
3.3 Upload Kode Program Ke ESP32	12
BAB IV TROUBLESHOOTING	24
4.1 LED Indikator pada ESP32	24
4.2 Kesalahan Umum dan Solusinya.....	24

DAFTAR GAMBAR

Gambar 1. Komponen Mobil Remote Kontrol	5
Gambar 2. Arduino IDE 1.8.19.....	6
Gambar 3. Dialog Box Administration	6
Gambar 4. Opsi Pilihan Instalasi	7
Gambar 5. Folder Penyimpanan	7
Gambar 6. Proses Instalasi.....	7
Gambar 7. Notifikasi Instalasi Driver.....	8
Gambar 8. Proses Instalasi Berhasil.....	8
Gambar 9. Tampilan Program Arduino IDE	8
Gambar 10. Menu File Arduino IDE	9
Gambar 11. Setting Additional Board.....	9
Gambar 12. Boards Manager Arduino IDE	10
Gambar 13. Pencarian Board ESP32	10
Gambar 14. Opsi Pilihan ESP32 Dev Module.....	10
Gambar 15. Tools Manage Libraries.....	11
Gambar 16. Library Manager	11
Gambar 17. Kode Deklarasi Library.....	12
Gambar 18. Kode Deklarasi PIN	12
Gambar 19. Kode Deklarasi Sensor.....	12
Gambar 20. Kode Koneksi ESP32, Wifi, dan MQTT.....	13
Gambar 21. Kode Pembacaan Sensor MPU6050	13
Gambar 22. Kode Pengiriman Topik Sensor MPU6050.....	14
Gambar 23. Kode Program GPS U-blox.....	15
Gambar 24. Kode Inialisasi Program Arduino.....	18
Gambar 25. Kode Eksekusi Program Arduino.....	18
Gambar 26. Kode Koneksi Wifi ke ESP32	19
Gambar 27. Kode Fungsi Callback.....	19
Gambar 28. Kode Fungsi Reconnect	20
Gambar 29. Verifikasi Kode Program.....	20
Gambar 30. Kode Program Python.....	21
Gambar 31. Tampilan Layar LCD	22

Gambar 32. Perintah Menjalankan Program.....	22
Gambar 33. Simulasi Kecelakaan Mobil Remot Kontrol	23
Gambar 34. Deteksi Pesan Alarm Pembacaan Sensor	23

DAFTAR TABEL

Tabel 1. Komponen Utama	1
-------------------------------	---

BAB I PENDAHULUAN

1.1 Gambaran Umum

Aplikasi ini dirancang untuk mendeteksi kecelakaan mobil dengan menggunakan sensor MPU6050 untuk mengukur percepatan, sensor GPS U-Blox untuk mendapatkan informasi lokasi, dan ESP32 sebagai mikrokontroler utama. Data hasil deteksi akan ditampilkan melalui OLED LCD dan akan mengirimkan sebuah alarm kecelakaan berupa pesan didalam telegram.

1.2 Komponen

Komponen dalam pembuatan dalam pembuatan aplikasi *crash detection* untuk mobil berbasis IoT sebagai berikut :

Tabel 1. Komponen Utama

No	Nama	Gambar	Jumlah	Keterangan
1.	Mikrokontroler ESP32		1	Untuk membaca dan menerima data sensor.
2.	Sensor MPU6050		1	Untuk mendeteksi accelerometer dan gyroscope.
3.	Sensor GPS Neo 6		1	Untuk mendeteksi titik lokasi.
4.	OLED LCD I2C 0.96 inci		1	Untuk menampilkan bacaan data.
5.	Project Board		1	Untuk merancang komponen.
6.	Kabel Jumper Female - female		25	Menghubungkan antara sensor dengan mikrokontroler atau project board.

No	Nama	Gambar	Jumlah	Keterangan
7.	Kabel Jumper Male - Male		25	Menghubungkan antara sensor dengan mikrokontroler atau project board.
8.	Mikro USB tipe B		1	Untuk mengupload kode ke dalam mikrokontroler
9.	Mobil Remote Control		1	Sebagai objek simulasi
10.	Solder		1	untuk menyambungkan dua material logam.
11.	Timah		1	menyambung beberapa lapisan perangkat yang membutuhkan kabel atau logam lain.
12.	Avometer		1	Mengukur arus listrik, tegangan listrik (AC dan DC), sekaligus resistensi.
13.	Baterai 2000 mAh		1	sumber energi yang dapat merubah energi kimia yang disimpannya menjadi energi listrik

1.3 Persyaratan Sistem

Dibutuhkan beberapa persyaratan sistem agar sistem dapat di install dan dijalankan sesuai prosedur.

1. Komputer sudah terinstal Arduino IDE.
2. VSCODE Untuk Mejalankan program python.
3. Kabel Jumper untuk menghubungkan komponen.
4. Sumber daya listrik untuk ESP32.
5. Mobil remote kontrol sebagai contoh simulasi yang akan digunakan.

BAB II PERSIAPAN HARDWARE

2.1 Koneksi ESP32 Ke Komputer

Pada tahapan ini pengguna diharapkan memiliki kabel usb tipe B atau micro usb sebagai syarat utama untuk menghubungkan perangkat ESP32 kedalam komputer dengan langkah sebagai berikut :

1. Siapkan kabel usb tipe B atau micro usb.
2. Pilih port dikomputer untuk menghubungkan.

2.2 Koneksi Sensor MPU6050

Hubungkan Sensor menggunakan kabel jumper dengan skema sebagai berikut :

1. Sambungkan VCC MPU6050 ke pin 3V3 pada ESP32.
2. Sambungkan GND MPU6050 ke GND ESP32.
3. Sambungkan pin SDA MPU6050 ke pin 21 SDA ESP32.
4. Sambungkan pin SCL MPU6050 ke pin 22 SCL ESP32.

2.3 Koneksi Sensor GPS U-Blox Neo6

Hubungkan Sensor menggunakan kabel jumper dengan skema sebagai berikut :

1. Sambungkan VCC GPS U-Blox ke pin 3V3 pada ESP32.
2. Sambungkan GND GPS U-Blox ke GND ESP32.
3. Sambungkan pin RX GPS U-Blox ke pin TX ESP32.
4. Sambungkan pin TX GPS U-Blox ke pin RX ESP32.

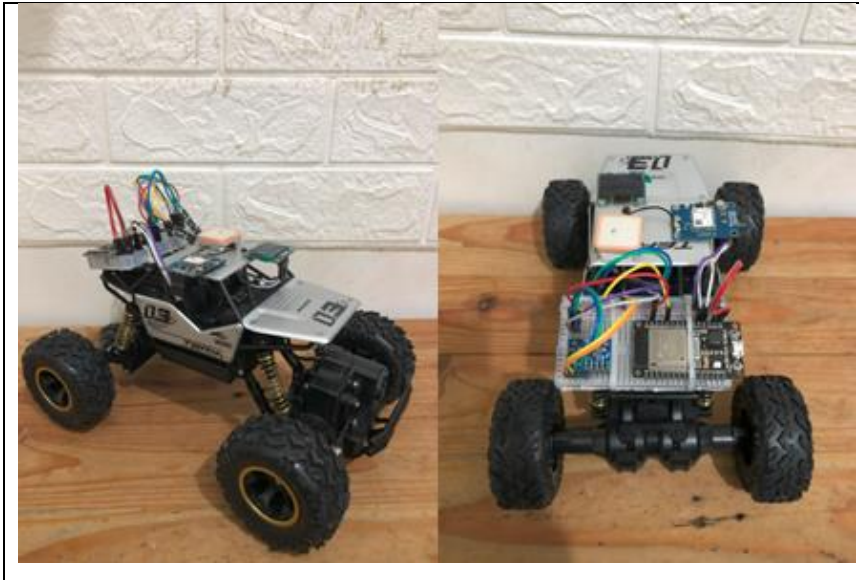
2.4 Koneksi OLED LCD

Hubungkan menggunakan kabel jumper dengan skema sebagai berikut :

1. Sambungkan VCC OLED LCD ke pin 3V3 pada ESP32.
2. Sambungkan GND OLED LCD ke GND ESP32.
3. Sambungkan pin SDA OLED LCD ke pin 21 SDA ESP32.
4. Sambungkan pin SCL OLED LCD ke pin 22 SCL ESP32.

2.5 Rangkai Komponen ke dalam Mobil Remote Kontrol

Setelah sensor – sensor tersambung ke ESP32 maka langkah selanjutnya, rangkai komponen tersebut kedalam mobile remote control seperti berikut :



Gambar 1. Komponen Mobil Remote Kontrol

Sesuaikan dan modifikasi dengan keadaan remote control yang anda punya, gunakan project board jika tidak memiliki solder untuk menyatukan kabel.

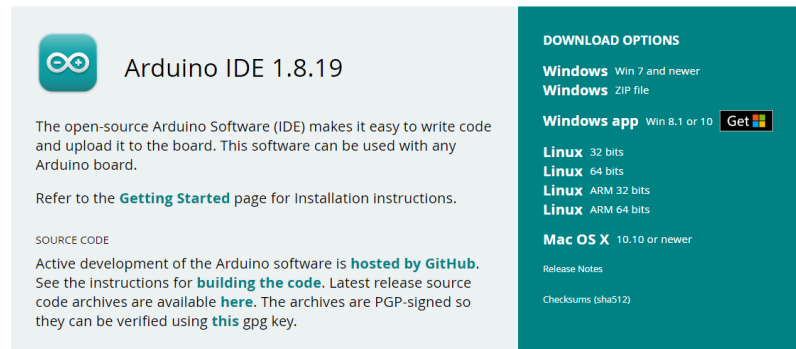
BAB III INSTALASI PERANGKAT LUNAK

3.1 Instalasi Arduino IDE

Pada tahapan ini adalah langkah – langkah penginstalan software Arduino IDE dan menambahkan ESP32 board.

1. Unduh Arduino IDE pada website berikut : <https://www.arduino.cc/en/software/>
2. Unduh Versi 1.8.19 untuk windows.

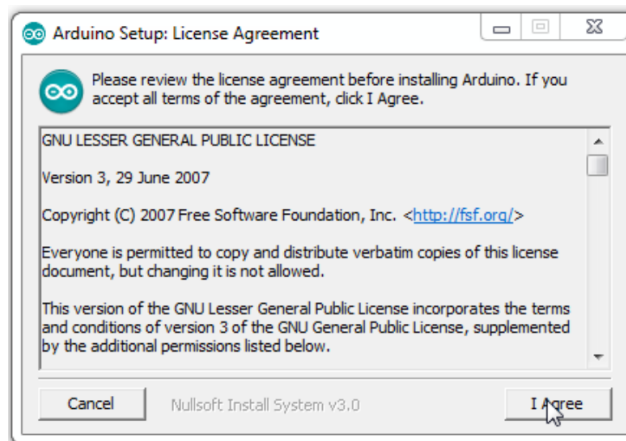
Legacy IDE (1.8.X)



Gambar 2. Arduino IDE 1.8.19

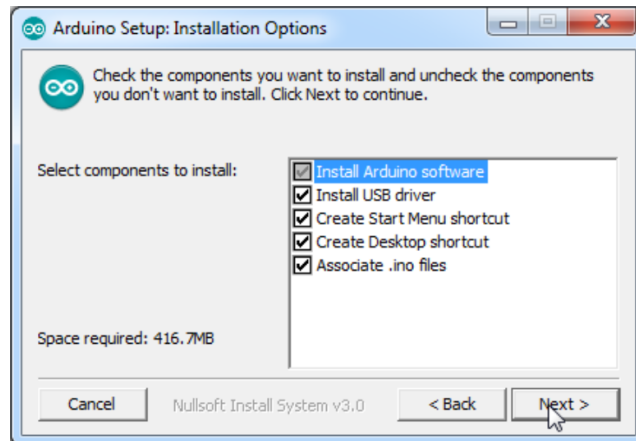
3. Langkah instalasi

- a. Setelah proses download selesai klik dua kali pada software Arduino IDE maka akan muncul dialog box administration klik yes, kemudian akan muncul kotak dialog selanjutnya sebagai berikut:



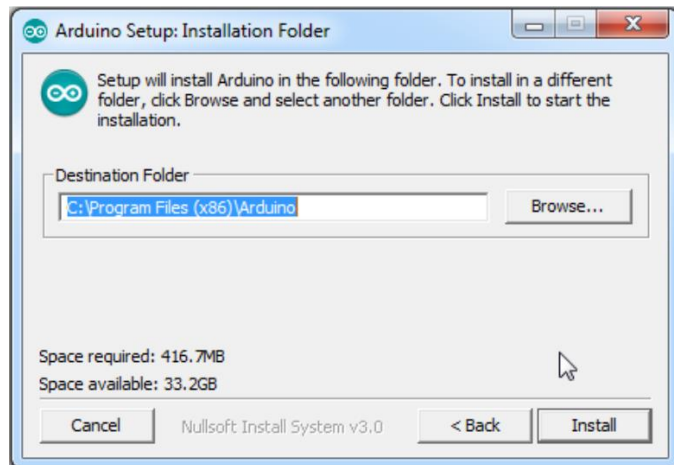
Gambar 3. Dialog Box Administration

- b. Pada dialog opsi instalasi pilih semua opsi kemudian klik tombol *next*.



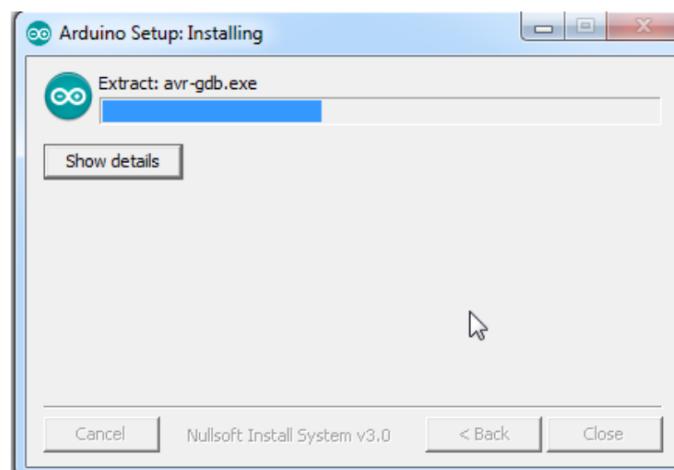
Gambar 4. Opsi Pilihan Instalasi

- c. Kemudian pilih folder untuk menyimpan Arduino IDE, klik tombol instal maka proses instalasi software akan dimulai.



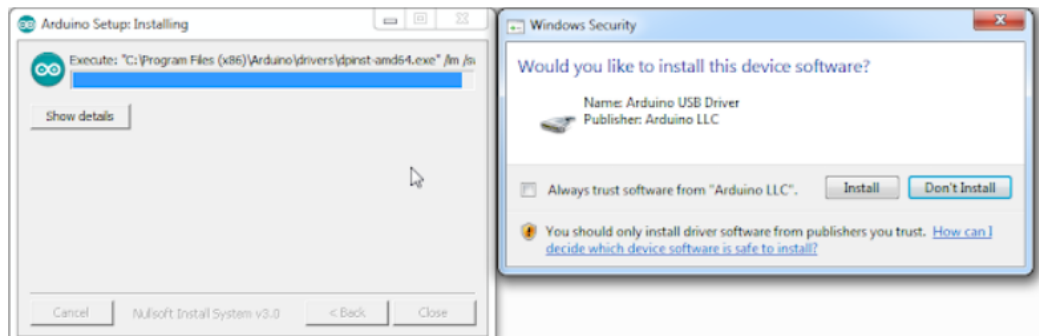
Gambar 5. Folder Penyimpanan

- d. Proses extract dan instalasi software memerlukan waktu beberapa menit.



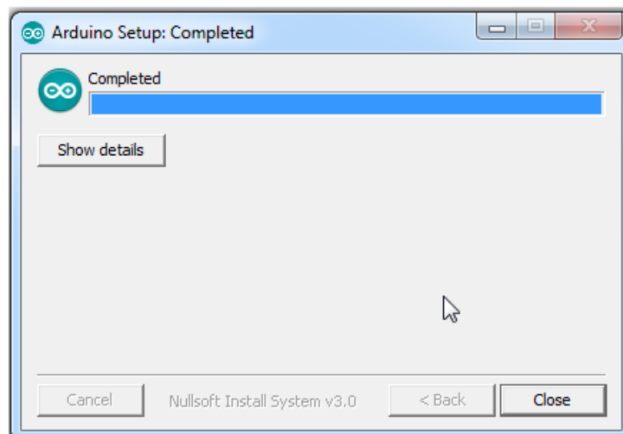
Gambar 6. Proses Instalasi

- e. Saat proses instalasi sedang berlangsung anda akan diminta untuk menginstal driver, pilih tombol instal proses ini bertujuan untuk mengenali dan melakukan komunikasi dengan board Arduino melalui port USB



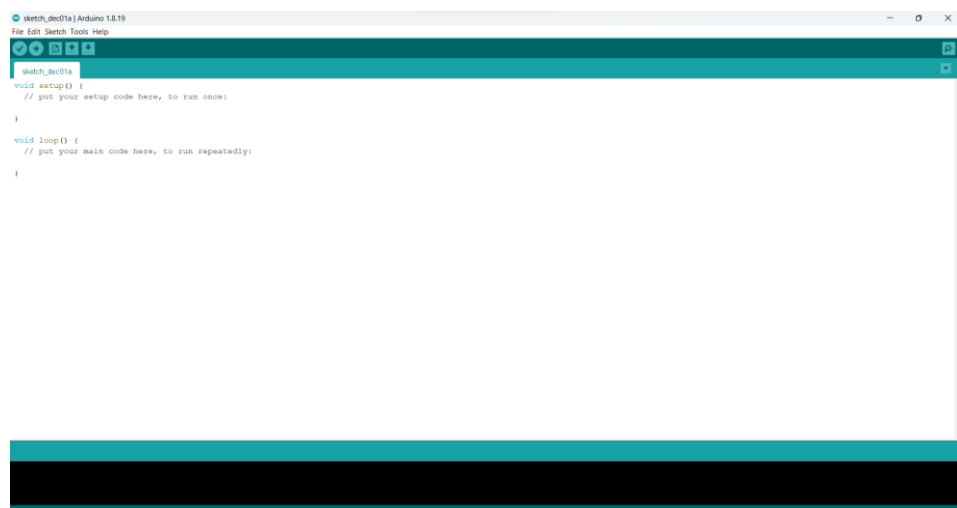
Gambar 7. Notifikasi Instalasi Driver

- f. Proses instalasi selesai.



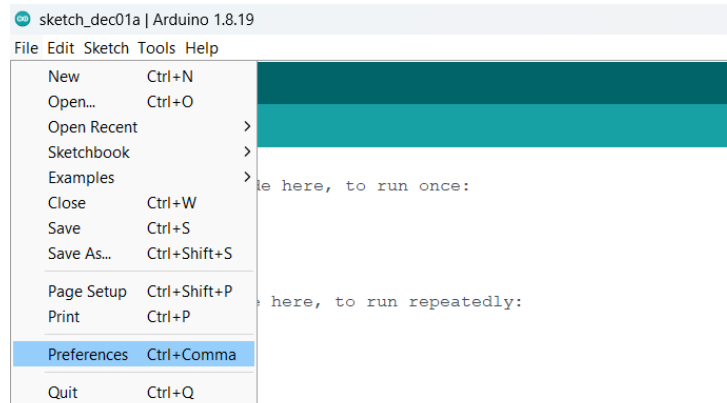
Gambar 8. Proses Instalasi Berhasil

- g. Setelah proses instalasi selesai klik dua kali pada desktop computer anda untuk menjalankan program Arduino IDE.



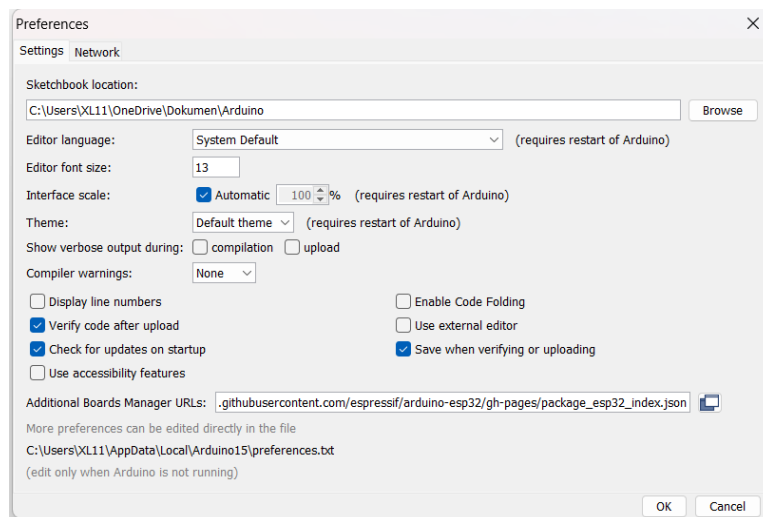
Gambar 9. Tampilan Program Arduino IDE

4. Tambahkan library ESP32 board kedalam Arduino IDE dengan cara sebagai berikut:
- Buka Arduino IDE kemudian klik menu file kemudian pilih preferensi



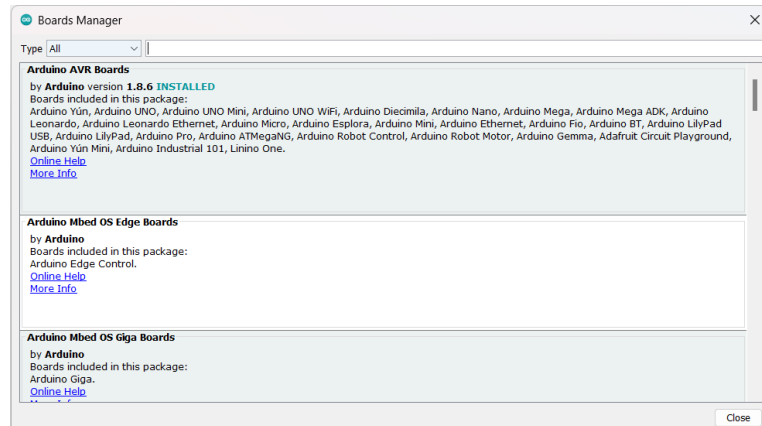
Gambar 10. Menu File Arduino IDE

- Tambahkan url berikut untuk menambahkan board ESP32 https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json pada kolom menu Additional Board Manager URLs, kemudian tekan OK



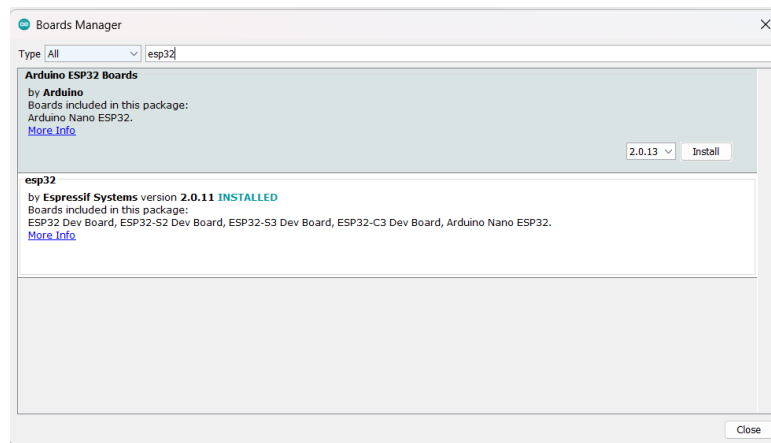
Gambar 11. Setting Additional Board

- c. Selanjutnya tambahkan instal board ESP32 dengan cara klik menu Tools kemudian pilih board, klik pada boards manager.



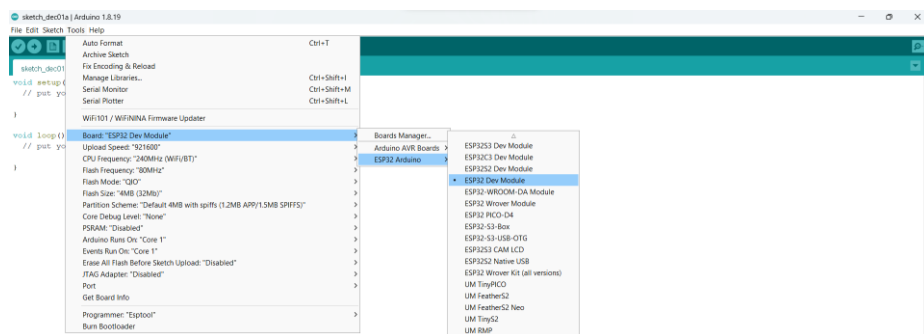
Gambar 12. Boards Manager Arduino IDE

- d. Pada menu board manager masukan esp32 kedalam box pencarian, pilih esp32 lalu klik install, kemudian tunggu sampai proses instalasi selesai.



Gambar 13. Pencarian Board ESP32

- e. Saat ingin menggunakan board esp32 masuk ke menu Tools kemudian pilih board ESP32 Arduino pilih ESP32 Dev Module.

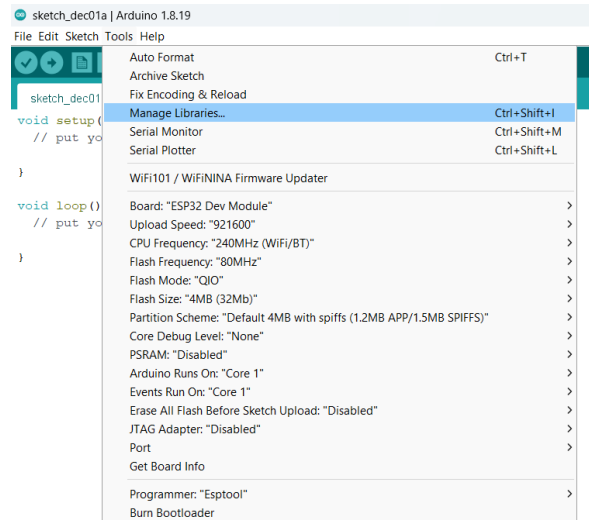


Gambar 14. Opsi Pilihan ESP32 Dev Module

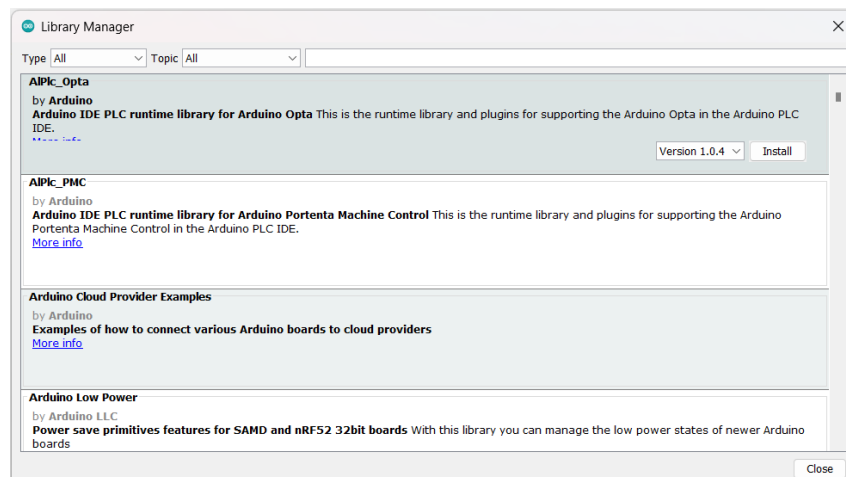
3.2 Instal Library Pada Arduino IDE

Pada tahapan ini adalah langkah – langkah menambahkan library dan depedensi untuk menjalankan sensor.

1. Buka Arduino IDE buat project baru kemudian pilih menu Tools pilih Manage Libraries.



Gambar 15. Tools Manage Libraries



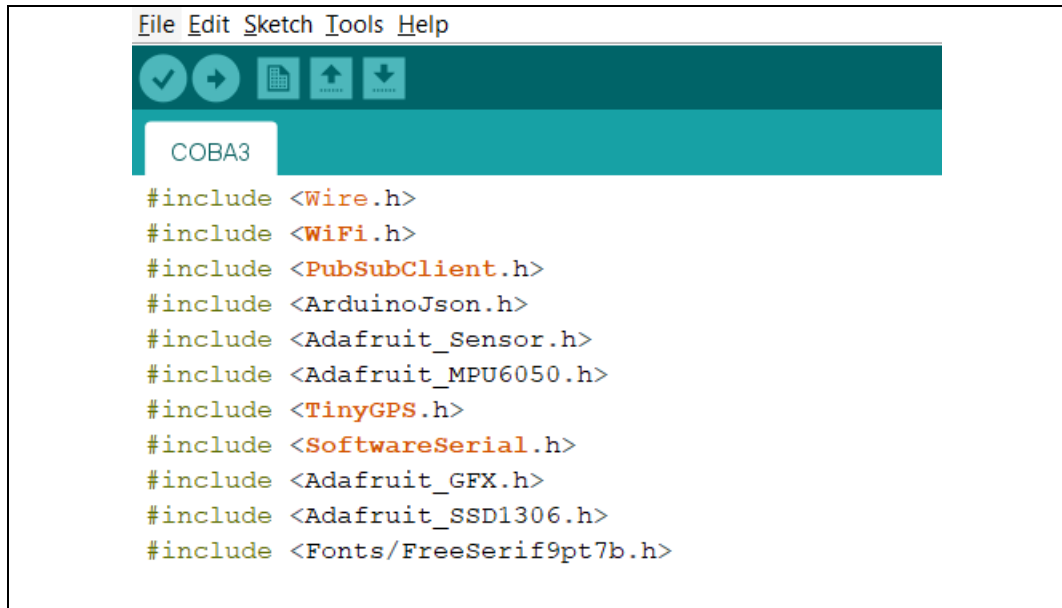
Gambar 16. Library Manager

2. Pada menu library manager cari dan install library berikut:
 - a. Adafruit MPU6050.
 - b. Adafruit SSD1306.
 - c. Adafruit GFX.
 - d. Arduino JSON.
 - e. PubSubClient.
 - f. TinyGPS.

3.3 Upload Kode Program Ke ESP32

Pada tahapan ini adalah memasukan kode program agar sensor - sensor dapat mendeteksi dan membaca data.

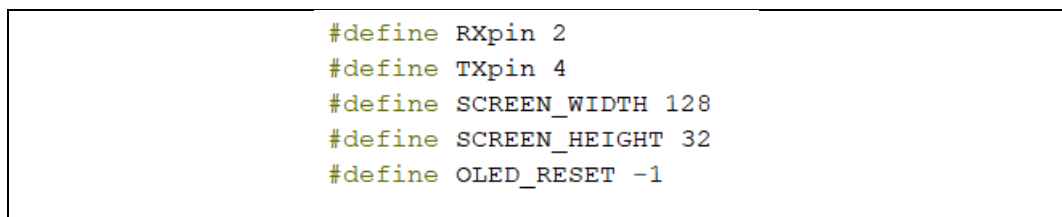
1. Langkah pertama deklarasikan library yang sudah diinstal di awal kode.



```
File Edit Sketch Tools Help
COBA3
#include <Wire.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_MPU6050.h>
#include <TinyGPS.h>
#include <SoftwareSerial.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Fonts/FreeSerif9pt7b.h>
```

Gambar 17. Kode Deklarasi Library

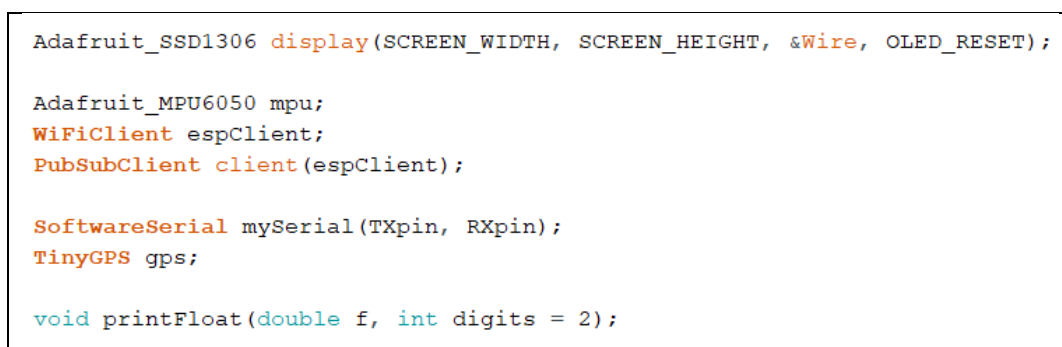
2. Selanjutnya deklarasikan pin untuk gps dan oled lcd



```
#define RXpin 2
#define TXpin 4
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 32
#define OLED_RESET -1
```

Gambar 18. Kode Deklarasi PIN

3. Deklarasi Sensor MPU6050, OLED LCD, GPS, Wifi, PubSubClient, dan Software Serial.



```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

Adafruit_MPU6050 mpu;
WiFiClient espClient;
PubSubClient client(espClient);

SoftwareSerial mySerial(TXpin, RXpin);
TinyGPS gps;

void printFloat(double f, int digits = 2);
```

Gambar 19. Kode Deklarasi Sensor

4. Kode perintah untuk menkoneksi ESP32 dengan Wifi serta topic MQTT.

```
const char *ssid = "JNT express 4"; // Replace with your WIFI SSID
const char *pass = "mabuknol"; // Replace with your WIFI PASSWORD
//const char *ssid = "JTI-POLINEMA"; // Replace with your WIFI SSID
//const char *pass = "jtifast!"; // Replace with your WIFI PASSWORD
const char* mqtt_server = "armadillo.rmcloudamqp.com";
const char* token = "sqvvgstbb:sqvvgstbb";
const char* passwordmqtt = "K0ygtSSHXEXxoJtFoPdS473MIqiY_AypF";
int port = 1883;

const char* topicdata = "trf/axis";

unsigned long lastMsg = 0;
#define MSG_BUFFER_SIZE (50)
char msg[MSG_BUFFER_SIZE];
int value = 0;
```

Gambar 20. Kode Koneksi ESP32, Wifi, dan MQTT

5. Kode untuk fungsi pembacaan sensor MPU6050

```
COBA3
void mpu_read() {
  sensors_event_t accel, gy;
  mpu.getAccelerometerSensor()->getEvent(&accel);
  mpu.getGyroSensor()->getEvent(&gy);

  float x = accel.acceleration.x;
  float y = accel.acceleration.y;

  float acc_magnitude = sqrt(x * x + y * y);
  display.setCursor(0, 1);
  display.print("Acc Total:");
  display.println(acc_magnitude);
  display.display();
  Serial.print(accel.acceleration.x);
  Serial.print(",");
  Serial.print(accel.acceleration.y);
  Serial.print(",");
  Serial.print(gy.gyro.x);
  Serial.print(",");
  Serial.print(gy.gyro.y);
  Serial.print(",");
  Serial.print("Acc total :");
  Serial.print(acc_magnitude);
  Serial.println("");
  if (acc_magnitude > 7) {
    Serial.println("Crash detected!");
    display.println("Crash detected!");
    display.display();
    delay(5000);
    display.clearDisplay();
  }
  delay(1000);
}
```

Gambar 21. Kode Pembacaan Sensor MPU6050

6. Kode untuk mengirimkan topic sensor MPU6050

```
String mpu_send_topic() {
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  float x = a.acceleration.x;
  float y = a.acceleration.y;
  float z = a.acceleration.z;
  float acc_magnitude = sqrt(x * x + y * y + z * z);

  String result = "";
  return result = "(" + String(a.acceleration.x) + "," + String(a.acceleration.y) + "," +
  String(a.acceleration.z) + "," + String(g.gyro.x) + "," +
  String(g.gyro.y) + "," + String(g.gyro.z) + "," + String(acc_magnitude) + ")";
}
```

Gambar 22. Kode Pengiriman Topik Sensor MPU6050

7. Kode program untuk GPS U-blox

```
void gpsdata(TinyGPS &gps)
{
  long latitude, longitude;
  float flatitude, flongitude;
  unsigned long date, time, chars, age;
  int year;
  byte month, day, hour, minute, second, hundredths;
  unsigned short sentences, failed;

  //latitude and longitude in 10^-5deg
  gps.get_position(&latitude, &longitude, &age);
  Serial.print("Lat/Long: "); Serial.print(latitude); Serial.print(", "); Serial.print(longitude);
  Serial.print(" Fix age: "); Serial.print(age); Serial.print("ms ");
  Serial.println();

  display.setCursor(0, 10);
  display.print("Lat: ");
  display.println(latitude);
  display.print("Long: ");
  display.println(longitude);
  display.display();
  delay(5000);
  display.clearDisplay();
}
```

Lanjutan Kode untuk GPS

```

void printFloat(double number, int digits)
{
    // Handle negative numbers
    if (number < 0.0)
    {
        Serial.print('-');
        number = -number;
    }

    // Round correctly so that print(1.999, 2) prints as "2.00"
    double rounding = 0.5;
    for (uint8_t i = 0; i < digits; ++i)
        rounding /= 10.0;

    number += rounding;

    // Extract the integer part of the number and print it
    unsigned long int_part = (unsigned long)number;
    double remainder = number - (double)int_part;
    Serial.print(int_part);

    // Print the decimal point, but only if there are digits beyond
    if (digits > 0)
        Serial.print(".");

    // Extract digits from the remainder one at a time
    while (digits-- > 0)
    {
        remainder *= 10.0;
        int toPrint = int(remainder);
        Serial.print(toPrint);
        remainder -= toPrint;
    }
}

```

Gambar 23. Kode Program GPS U-blox

8. Setup kode sebagai bentuk inisialisasi program Arduino dan hanya dieksekusi sekali sejak program dijalankan.

```
COBA3 §  
  
void setup() {  
  Serial.begin(9600);  
  setup_wifi();  
  Wire.begin();  
  mySerial.begin(9600);  
  Serial.println("GPS Mulai");  
  while (!Serial) {}  
  
  if (!mpu.begin()) {  
    Serial.println("Failed to find MPU6050 chip");  
    while (1) {  
      delay(10);  
    }  
  }  
  
  mpu.setAccelerometerRange(MPU6050_RANGE_16_G);  
  Serial.print("Accelerometer range set to: ");  
  switch (mpu.getAccelerometerRange()) {  
    case MPU6050_RANGE_2_G:  
      Serial.println("+2G");  
      break;  
    case MPU6050_RANGE_4_G:  
      Serial.println("+4G");  
      break;  
    case MPU6050_RANGE_8_G:  
      Serial.println("+8G");  
      break;  
    case MPU6050_RANGE_16_G:  
      Serial.println("+16G");  
      break;  
  }  
}
```

Lanjutan Kode

```

mpu.setGyroRange(MPU6050_RANGE_250_DEG);
Serial.print("Gyro range set to: ");
switch (mpu.getGyroRange()) {
  case MPU6050_RANGE_250_DEG:
    Serial.println("+ 250 deg/s");
    break;
  case MPU6050_RANGE_500_DEG:
    Serial.println("+ 500 deg/s");
    break;
  case MPU6050_RANGE_1000_DEG:
    Serial.println("+ 1000 deg/s");
    break;
  case MPU6050_RANGE_2000_DEG:
    Serial.println("+ 2000 deg/s");
    break;
}

```

Lanjutan Kode

```

mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
Serial.print("Filter bandwidth set to: ");
switch (mpu.getFilterBandwidth()) {
  case MPU6050_BAND_260_HZ:
    Serial.println("260 Hz");
    break;
  case MPU6050_BAND_184_HZ:
    Serial.println("184 Hz");
    break;
  case MPU6050_BAND_94_HZ:
    Serial.println("94 Hz");
    break;
  case MPU6050_BAND_44_HZ:
    Serial.println("44 Hz");
    break;
  case MPU6050_BAND_21_HZ:
    Serial.println("21 Hz");
    break;
  case MPU6050_BAND_10_HZ:
    Serial.println("10 Hz");
    break;
  case MPU6050_BAND_5_HZ:
    Serial.println("5 Hz");
    break;
}

```

Lanjutan Kode

```

display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);

client.setServer(mqtt_server, port);
client.setCallback(callback);
client.subscribe(topicdata);
}

```

Gambar 24. Kode Inialisasi Program Arduino

9. Kode program untuk void loop yang bertujuan untuk mengeksekusi dan menjalankan program.

```

void loop() {
    bool newdata = false;
    unsigned long start = millis();

    while (millis() - start < 1000)
    {
        if (mySerial.available())
        {
            char c = mySerial.read();
            if (gps.encode(c))
            {
                newdata = true;
            }
        }
    }

    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    unsigned long now = millis();
    if (now - lastMsg > 1000) { //perintah publish data
        client.publish(topicdata, mpu_send_topic().c_str());
        lastMsg = now;
        mpu_read();
        //Serial.println(mpu_send_topic());
    }

    if (newdata)
    {
        //    gpsdata(gps);
    }
}

```

Gambar 25. Kode Ekesekusi Program Arduino

10. Fungsi setup wifi perintah untuk mengkoneksikan ESP32 kedalam jaringan Wifi

```
void setup_wifi() { //perintah koneksi wifi
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA); //setting wifi chip sebagai station/client
  WiFi.begin(ssid, pass); //koneksi ke jaringan wifi

  while (WiFi.status() != WL_CONNECTED) { //perintah tunggu esp32 sampai
    delay(500);
    Serial.print(".");
  }

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

Gambar 26. Kode Koneksi Wifi ke ESP32

11. Void callback perintah untuk menampilkan data ketika esp32 di setting sebagai subscriber.

```
void callback(char* topic, byte * payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) { //mengecek jumlah data yang a
    Serial.print((char)payload[i]);
  }
  Serial.println();
  if ((char)payload[0] == '1') {
    digitalWrite(2, LOW);
  } else {
    digitalWrite(2, HIGH);
  }
}
```

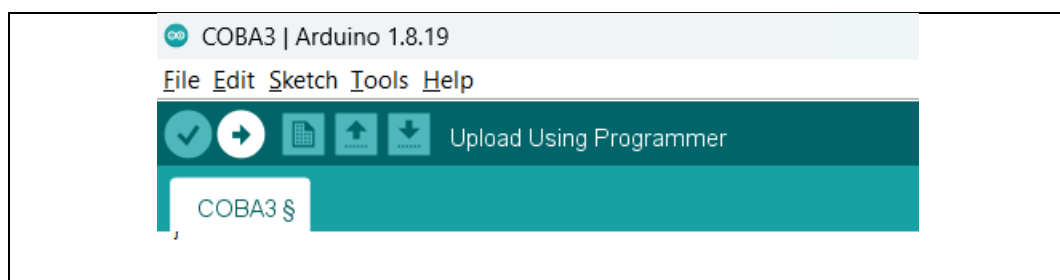
Gambar 27. Kode Fungsi Callback

12. Void reconnect perintah koneksi ESP32 ke mqtt broker baik itu sebagai publisher atau subscriber.

```
void reconnect() { //perintah koneksi esp32 ke mqtt broker baik itu
// Loop until we're reconnected
while (!client.connected()) {
  Serial.print("Attempting MQTT connection...");
  // perintah membuat client id agar mqtt broker mengenali board :
  String clientId = "ESP32Client-";
  clientId += String(random(0xffff), HEX);
  // Attempt to connect
  if (client.connect(clientId.c_str(), token, passwordmqtt)) {
    Serial.println("Connected");
  } else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    delay(5000);
  }
}
}
```

Gambar 28. Kode Fungsi Reconnect

13. Setelah kode selesai di buat upload kode tersebut kedalam ESP32 dengan cara klik tombol panah kanan diatas dan jangan lupa untuk memverifikasi terlebih dahulu dengan mengklik tombol centang untuk mengetahui kode error atau tidak.



Gambar 29. Verifikasi Kode Program

14. Untuk kode lengkap silahkan lihat pada halaman berikut :
<https://drive.google.com/drive/folders/1ghjkcGM14ogYU8aYONHyMQ9slbynNpBk?usp=sharing/>

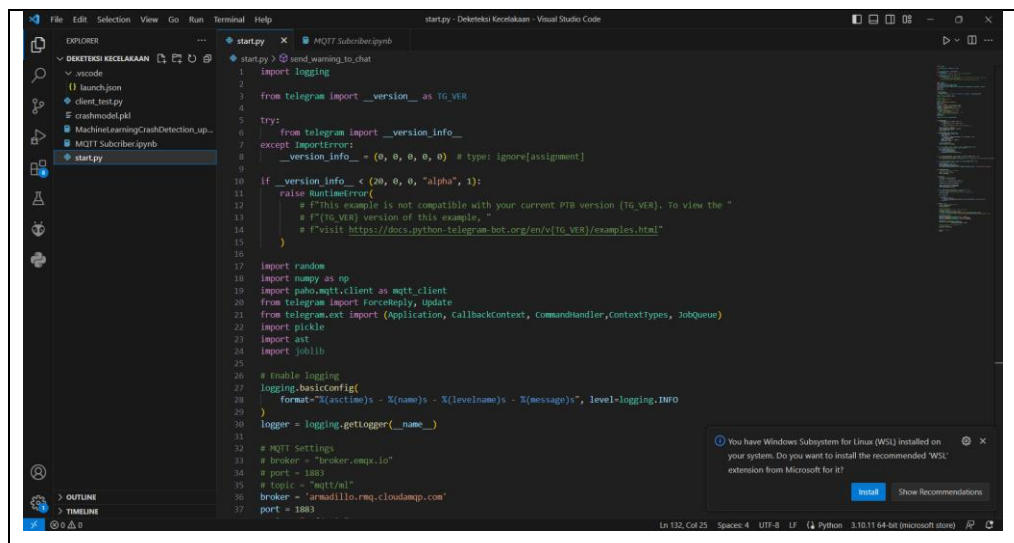
3.4 Menjalankan program python sebagai server pengiriman data kecelakaan

Setelah mengupload kode kedalam ESP32 langkah selanjutnya menjalankan program python sebagai server untuk mengirimkan pesan alarm ke telegram

1. Unduh file python pada link berikut :

https://drive.google.com/drive/folders/1L_KbFJWh1wQNulKtu6gO4JgeRWA_CVr_h?usp=sharing/

2. Setelah file di unduh kemudia buka pada aplikasi vscode, untuk langkah penginstalan aplikasi vscode dapat dilihat pada link berikut : https://education-launchcode-org.translate.goog/lchs/appendices/software-installation/vscode-setup.html?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc/
3. Klik pada file start.py jangan lupa install terlebih dahulu ekstensi python didalam vscode agar tidak terjadi error.



```
1 import logging
2
3 from telegram import __version__ as TG_VER
4
5 try:
6     from telegram import __version_info__
7 except ImportError:
8     __version_info__ = (0, 0, 0, 0) # type: ignore[assignment]
9
10 if __version_info__ < (20, 0, 0, "alpha", 1):
11     raise RuntimeError(
12         # f"This example is not compatible with your current PyT version {TG_VER}. To view the "
13         # f"({TG_VER}) version of this example, "
14         # f"visit https://docs.python-telegram-bot.org/en/{TG_VER}/examples.html"
15     )
16
17 import random
18 import numpy as np
19 import paho.mqtt.client as mqtt
20 from telegram import ForceReply, Update
21 from telegram.ext import (Application, CallbackContext, CommandHandler, ContextTypes, JobQueue)
22 import pickle
23 import ast
24 import joblib
25
26 # Enable logging
27 logging.basicConfig(
28     format='%(asctime)s - %(name)s - %(levelname)s - %(message)s', level=logging.INFO
29 )
30 logger = logging.getLogger(__name__)
31
32 # MQTT Settings
33 # broker = "broker.emqx.io"
34 # port = 1883
35 # topic = "mqtt/a"
36 broker = "aradillo.rmq.cloudamp.com"
37 port = 1883
```

Gambar 30. Kode Program Python

4. Eksekusi program python dengan cara mengklik tombol play pada pojok kanan atas pada software vscode jika berhasil maka program akan siap menerima bacaan data dari sensor MPU6050 untuk mendeteksi kecelakaan.

3.5 Melakukan Simulasi kecelakaan

Untuk melakukan simulasi kecelakaan langkah pertama adalah beri daya pada esp32 dan mobil remote control, setelah itu ikuti langkah berikut:

1. Hidupkan saklar mobil remote yang dimana dayanya sudah menyatu dengan ESP32.
2. Tunggu sampai LCD dan ESP32 terkoneksi dengan wifi anda dapat melihatnya pada pesan di terminal yang di jalankan pada program python.
3. Jika sudah terkoneksi lihat pada bagian layar LCD akan menunjukkan akselarsi total dan juga posisi longitude dan latitude seperti gambar dibawah ini:



Gambar 31. Tampilan Layar LCD

4. Buka telegram kemudian subscribe channel berikut https://t.me/crash_alarm_bot
5. Kemudian masukan perintah /start untuk memulai bot dan masukan perintah /subscribe untuk mensubscribe data yang akan di kirimkan oleh sensor MPU6050 melalui program python.



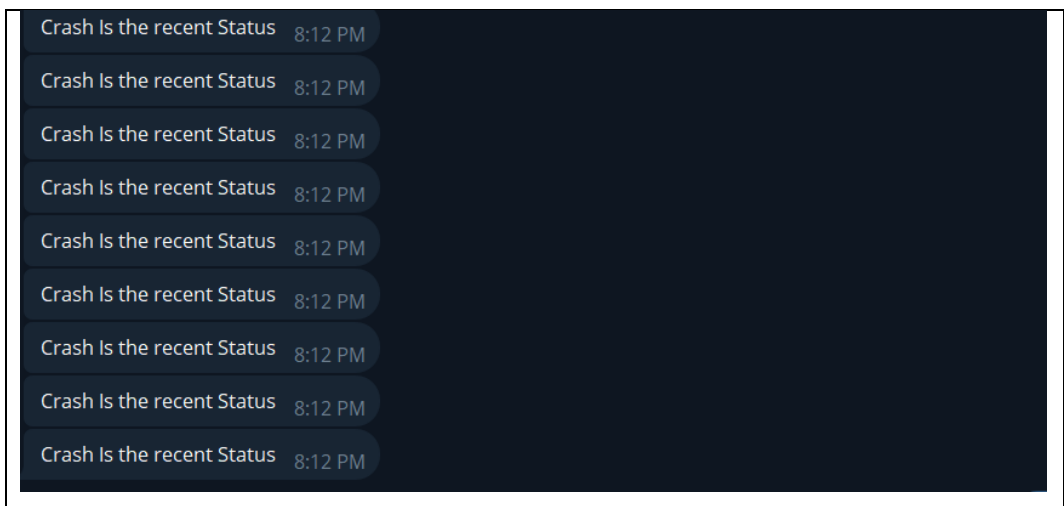
Gambar 32. Perintah Menjalankan Program

- Langkah selanjutnya lakukan simulasi kecelakaan dengan cara mobil remote kontrol ditabrakan ke arah dinding dengan jarak yang cukup jauh.



Gambar 33. Simulasi Kecelakaan Mobil Remot Kontrol

- Jika berhasil mendeteksi kecelakaan maka bot telegram akan mengirimkan sebuah pesan alarm yang dimana mendeteksi status terakhir dari pembacaan sensor tersebut.



Gambar 34. Deteksi Pesan Alarm Pembacaan Sensor

BAB IV TROUBLESHOOTING

4.1 LED Indikator pada ESP32

1. Jika LED indikator berkedip cepat, terjadi kesalahan koneksi.
2. Periksa kembali semua koneksi hardware.

4.2 Kesalahan Umum dan Solusinya

1. Jika LED indikator berkedip cepat, terjadi kesalahan koneksi.
2. Periksa kembali semua koneksi hardware.
3. Jika GPS tidak mendapatkan sinyal bawa ketempat terbuka.