

SOP Alat Pendeteksi Banjir Otomatis Berbasis Arduino dan Sensor Ultrasonik

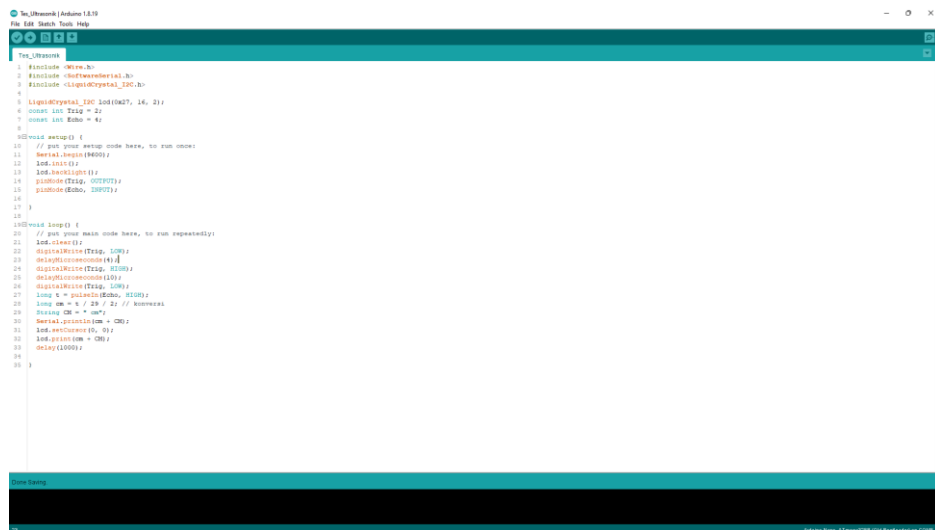
Langkah 1: Membaca Doa

Langkah 2: Persiapkan Arduino Nano

- Lepaskan Arduino Nano dari kotak panel.

Langkah 3: Persiapkan Perangkat Lunak

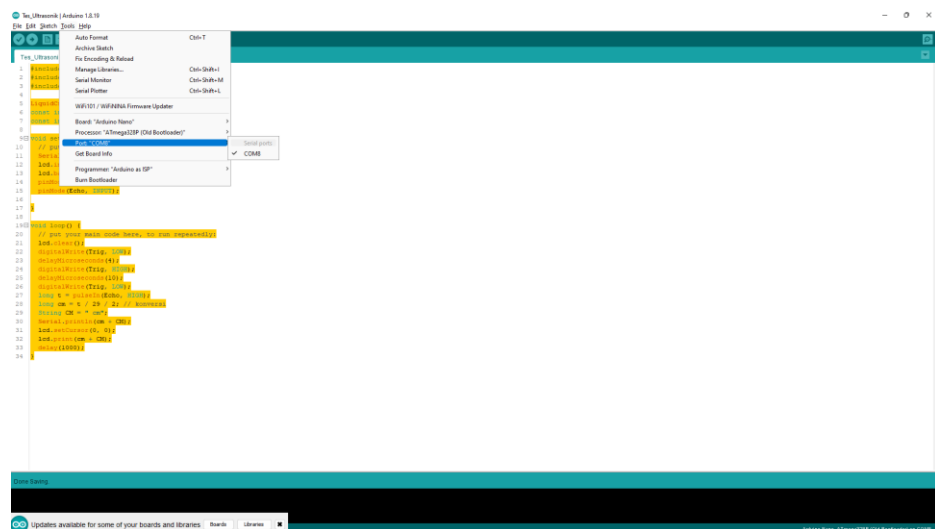
- Hubungkan Arduino ke laptop dan buka program "Tes Ultrasonik" (Lampiran 1).



```
1 #include <Wire.h>
2 #include <SoftwareSerial.h>
3 #include <LiquidCrystal_I2C.h>
4
5 LiquidCrystal_I2C lcd(0x27, 16, 2);
6 const int Trig = 2;
7 const int Echo = 4;
8
9
10 void setup() {
11   // put your setup code here, to run once:
12   Serial.begin(9600);
13   lcd.begin(16,2);
14   pinMode(Trig, OUTPUT);
15   pinMode(Echo, INPUT);
16 }
17
18
19 void loop() {
20   // put your main code here, to run repeatedly:
21   lcd.clear();
22   digitalWrite(Trig, LOW);
23   delayMicroseconds(5);
24   digitalWrite(Trig, HIGH);
25   delayMicroseconds(10);
26   digitalWrite(Trig, LOW);
27   long t = pulseIn(Echo, HIGH);
28   long cm = t / 29 / 2; // convert
29   Serial.print(cm);
30   Serial.println(" cm");
31   lcd.setCursor(0, 0);
32   lcd.print(cm);
33   delay(1000);
34 }
35 }
```

Langkah 4: Konfigurasi Program

- Sebelum mengunggah, ubah pengaturan board menjadi Arduino Nano dan pilih port yang tersedia di menu tools.



```
1 #include <Wire.h>
2 #include <SoftwareSerial.h>
3 #include <LiquidCrystal_I2C.h>
4
5 LiquidCrystal_I2C lcd(0x27, 16, 2);
6 const int Trig = 2;
7 const int Echo = 4;
8
9
10 void setup() {
11   // put your setup code here, to run once:
12   Serial.begin(9600);
13   lcd.begin(16,2);
14   pinMode(Trig, OUTPUT);
15   pinMode(Echo, INPUT);
16 }
17
18
19 void loop() {
20   // put your main code here, to run repeatedly:
21   lcd.clear();
22   digitalWrite(Trig, LOW);
23   delayMicroseconds(5);
24   digitalWrite(Trig, HIGH);
25   delayMicroseconds(10);
26   digitalWrite(Trig, LOW);
27   long t = pulseIn(Echo, HIGH);
28   long cm = t / 29 / 2; // convert
29   Serial.print(cm);
30   Serial.println(" cm");
31   lcd.setCursor(0, 0);
32   lcd.print(cm);
33   delay(1000);
34 }
```

Langkah 5: Unggah Kode

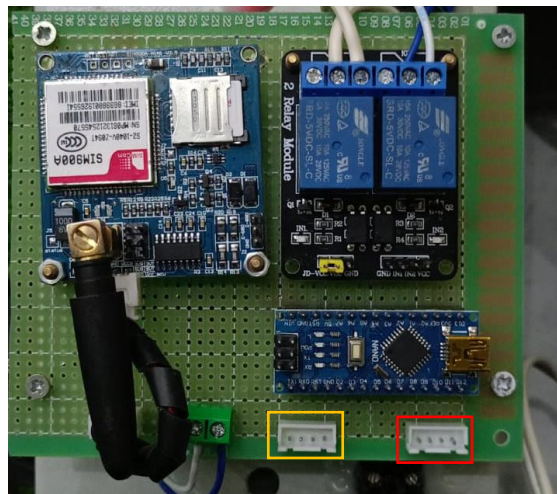
- Unggah kode hingga selesai.

Langkah 6: Kembali Pasang Arduino

- Cabut Arduino dari laptop dan pasang kembali Arduino Nano ke posisinya di kotak panel.

Langkah 7: Pemasangan Sensor dan LCD

- Pasang Sensor Ultrasonik dan LCD pada tempat yang telah disediakan (pastikan tidak terpasang terbalik untuk menghindari kerusakan).



Keterangan:

Kotak Kuning = LCD

Kotak Merah = Ultrasonik

Langkah 8: Aktivasi Perangkat

- Hidupkan Arduino dengan daya AC dan tetap tempatkan pada kotak panel tanpa dilepaskan.

Langkah 9: Pengecekan Awal

- Lihat pada LCD apakah mengeluarkan output jarak atau tidak. Jika tidak, periksa kabel penghubung antara sensor dan alat.

Langkah 10: Pemasangan Sensor Ultrasonik

- Pasang sensor ultrasonik pada pipa yang sudah disediakan, pastikan lurus, dan output dari sensor ≥ 85 cm.

Langkah 11: Pengujian

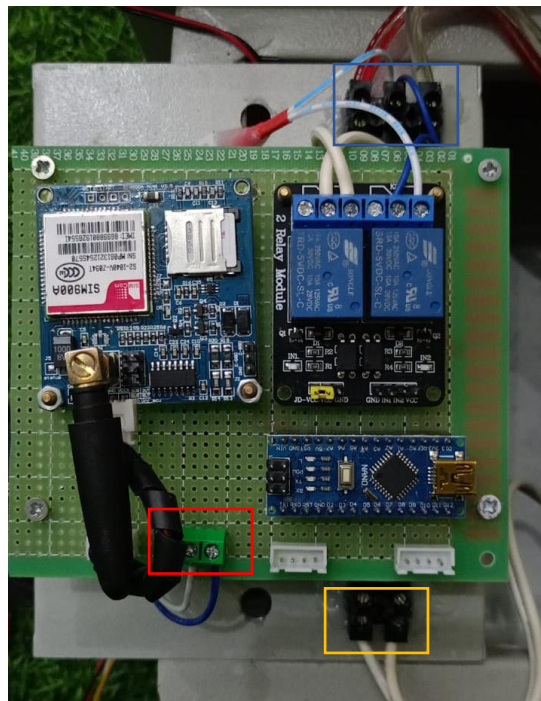
- Setelah pengujian selesai, matikan sumber daya dari kotak panel, dan cabut kembali Arduino dari tempatnya.

Langkah 12: Pengembalian Program Asli

- Unggah program asli dari alat tersebut. Setelah berhasil diunggah, lepaskan USB dari Arduino (Lampiran 2).

Langkah 13: Pemeriksaan Internal

- Pastikan semua kabel di dalam kotak panel terpasang dengan kuat dan baik.



Keterangan:

Kotak Kuning = Penghubung Sumber AC

Kotak Merah = Penghubung Step Down 5V ke Arduino

Kotak Biru = Penghubung Lampu

Langkah 14: Pemeriksaan Sensor

- Sebelum menyalakan, periksa posisi sensor ultrasonik untuk memastikan tidak terhalang oleh objek apa pun yang dapat menyebabkan alarm berbunyi secara tidak perlu.

Langkah 15: Aktivasi Perangkat

- Nyalakan perangkat dengan daya AC.

Lampiran 1.

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

```
#include <SoftwareSerial.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
const int Trig = 2;
```

```
const int Echo = 4;
```

```
void setup() {
```

```
    // put your setup code here, to run once:
```

```
    Serial.begin(9600);
```

```
    lcd.init();
```

```
    lcd.backlight();
```

```
    pinMode(Trig, OUTPUT);
```

```
    pinMode(Echo, INPUT);
```

```
}
```

```
void loop() {
```

```
    // put your main code here, to run repeatedly:
```

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

```
    digitalWrite(Trig, LOW);
```

```
    delayMicroseconds(4);
```

```
digitalWrite(Trig, HIGH);  
  
delayMicroseconds(10);  
  
digitalWrite(Trig, LOW);  
  
long t = pulseIn(Echo, HIGH);  
  
long cm = t / 29 / 2; // konversi  
  
String CM = " cm";  
  
Serial.println(cm + CM);  
  
lcd.setCursor(0, 0);  
  
lcd.print(cm + CM);  
  
delay(1000);  
  
}
```

Lampiran 2.

/*

Pin Pin:

Ultrasonik:

Trig=D2

Echo=D4

Relay:

ch1=5

ch2=6

Sim900a:

rx=11

tx=10

*/

#include <SoftwareSerial.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

char msg;

char call;

```
const int relay1 = 5;
```

```
const int relay2 = 6;
```

```
const int Trig = 2;
```

```
const int Echo = 4;
```

```
const int rx = 9;
```

```
const int tx = 10;
```

```
int relayON = LOW;
```

```
int relayOFF = HIGH;
```

```
SoftwareSerial GSM(11, 10);
```

```
char phone_no[] = "+6281936087763"; //change with phone number to phone
```

```
// Tentukan status untuk tampilan LCD
```

```
enum DisplayState {
```

```
    DEVICE_NAME,
```

```
    FLOOD_STATUS
```

```
};
```

```
// Inisialisasi status tampilan
```

```
DisplayState displayState = DEVICE_NAME;
```

```
// Tanda untuk melacak tindakan SMS dan panggilan
```



```
bool smsSent = false;

bool callMade = false;


void setup() {

    Serial.begin(9600);

    GSM.begin(9600);


    lcd.init();

    lcd.backlight();


    pinMode(relay1, OUTPUT);

    pinMode(relay2, OUTPUT);

    pinMode(Trig, OUTPUT);

    pinMode(Echo, INPUT);


    digitalWrite(relay1, relayOFF);

    digitalWrite(relay2, relayOFF);


    Serial.println("Initializing....");

    initModule("AT", "OK", 1000);

}


void loop() {

    lcd.clear();
```

```

// Ukur dan tampilkan status banjir

digitalWrite(Trig, LOW);

delayMicroseconds(4);

digitalWrite(Trig, HIGH);

delayMicroseconds(10);

digitalWrite(Trig, LOW);

long t = pulseIn(Echo, HIGH);

long cm = t / 29 / 2; // konversi

String CM = " cm";

Serial.println(cm + CM);


switch (displayState) {

    case DEVICE_NAME:

        // Tampilkan nama perangkat

        lcd.setCursor(0, 0);

        lcd.print(" Fadelnadana");

        lcd.setCursor(0, 1);

        lcd.print(" Flood Detector");

        delay(1000); // Tunggu 1 detik

        displayState = FLOOD_STATUS; // Pindah ke status berikutnya

        break;


    case FLOOD_STATUS:

        if (cm <= 30) {

```

```

digitalWrite(relay1, relayON);

digitalWrite(relay2, relayON);

Serial.println("Relay Nyala");

lcd.setCursor(0, 0);

lcd.print("Status Banjir:");

lcd.setCursor(0, 1);

lcd.print("AWAS !! ");

if (!callMade) {

    SendMessageAwasi1();

    //SendMessageAwasi2();

    //callUp(phone_no);

    Serial.println("TELPON AWAS !!");

    callMade = true; // Setel tanda menjadi true setelah panggilan dibuat

}

} else if (cm <= 70 && cm > 30) {

    digitalWrite(relay1, relayOFF);

    digitalWrite(relay2, relayOFF);

    lcd.setCursor(0, 0);

    lcd.print("Status Banjir:");

    lcd.setCursor(0, 1);

    lcd.print("SIAGA !! ");

    if (!smsSent) {

        SendMessageSiaga1();

        SendMessageSiaga2();

```

```

        Serial.println("SMS SIAGA !!");

        smsSent = true; // Setel tanda menjadi true setelah SMS dikirim
    }

    } else {

        digitalWrite(relay1, relayOFF);

        digitalWrite(relay2, relayOFF);

        if (smsSent || callMade) {

            smsSent = false; // Reset tanda SMS saat status banjir berubah

            callMade = false; // Reset tanda panggilan saat status banjir berubah

        }

        lcd.setCursor(0, 0);

        lcd.print("Status Banjir:");

        lcd.setCursor(0, 1);

        lcd.print("AMAN");

    }

    delay(1000); // Tunggu 1 detik

    displayState = DEVICE_NAME; // Pindah kembali ke status nama perangkat

    break;

}

}

void initModule(String cmd, char *res, int t) {

    while (1) {

```

```

Serial.println(cmd);

GSM.println(cmd);

delay(100);

while (GSM.available() > 0) {

    if (GSM.find(res)) {

        Serial.println(res);

        delay(t);

        return;

    } else {

        Serial.println("Error");

    }

}

delay(t);

}

}

void callUp(char *number) {

    GSM.print("ATD + "); GSM.print(number); GSM.println(";"); //Call to the specific number,
ends with semi-colon,replace X with mobile number

    delay(20000);    // wait for 20 seconds...

    GSM.println("ATH"); //hang up

    delay(100);

}

void SendMessageSiaga1()

```

```

{

GSM.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1000 milli seconds or 1 second

GSM.println("AT+CMGS=\"+6281936087763\"\\r"); // Replace x with mobile number

delay(1000);

GSM.println("Peringatan Banjir! Ketinggian air adalah 30 cm, kondisi siaga. Air mulai naik,
harap tetap waspada");// The SMS text you want to send

delay(100);

GSM.println((char)26);// ASCII code of CTRL+Z

delay(1000);

}

```

void SendMessageSiaga2()

```

{

GSM.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

delay(1000); // Delay of 1000 milli seconds or 1 second

GSM.println("AT+CMGS=\"+6281936087763\"\\r"); // Replace x with mobile number

delay(1000);

GSM.println("Pastikan persediaan darurat Anda. Ikuti petunjuk evakuasi jika diperlukan!");//
The SMS text you want to send

delay(100);

GSM.println((char)26);// ASCII code of CTRL+Z

delay(1000);

}

```

```

void SendMessageAwasi1()

{

    GSM.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

    delay(1000); // Delay of 1000 milli seconds or 1 second

    GSM.println("AT+CMGS=\"+6281936087763\"\\r"); // Replace x with mobile number

    delay(1000);

    GSM.println("Peringatan Banjir! Ketinggian air adalah 70 cm, kondisi darurat. Waspada!
    kenaikan air. Segera pindahkan diri ke tempat yang lebih tinggi."); // The SMS text you want
    to send

    delay(100);

    GSM.println((char)26); // ASCII code of CTRL+Z

    delay(1000);

}

```

```

void SendMessageAwasi2()

{

    GSM.println("AT+CMGF=1"); //Sets the GSM Module in Text Mode

    delay(1000); // Delay of 1000 milli seconds or 1 second

    GSM.println("AT+CMGS=\"+6281936087763\"\\r"); // Replace x with mobile number

    delay(1000);

    GSM.println("Ikuti arahan dan pantau perkembangan melalui sumber informasi resmi!
    #KeselamatanAndaPrioritas"); // The SMS text you want to send

    delay(100);

    GSM.println((char)26); // ASCII code of CTRL+Z

    delay(1000);

}

```

```
}
```

```
void MakeCall()
```

```
{
```

```
    GSM.println("ATD+6281936087763;"); // ATDxxxxxxxxxx; -- watch out here for semicolon  
    at the end!!
```

```
    Serial.println("Calling "); // print response over serial port
```

```
    delay(1000);
```

```
}
```