LAMPIRAN PROGRAM

```
#include <WiFi.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <LiquidCrystal_I2C.h>

// Inisialisasi LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);

unsigned long lastUpdate;
unsigned long lcdSwitch;
#define periodLcdSwitch 2000
bool lcdToggle = 0;

bool isAlarmActive = false;
bool isReset = false;

// Pin Konfigurasi
#define TRIG_PIN 26  // GPIO 18
#define ECHO_PIN 25  // GPIO 19
#define BUTTON_PIN 12  // GPIO 0
#define BUZZER_PIN 13  // GPIO 13
#define DOOR_PIN 15  // Pin for magnetic door reed switch

// Konfigurasi WiFi
const char* ssid = "ikoo"; //"ikoo";
const char* password = "0987654321";// "0987654321";

// NTP Konfigurasi
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

// Variabel untuk menyimpan tanggal dan waktu
String Time;
String jamStamp;
String menitStamp;
String detikStamp;
int jam, menit, detik;

// Deklarasi Variabel
int tinggiBotol = 23; // 230 mm atau 23.0 cm, misalnya
int kebutuhan = 2300; // 2300 mililiter
int wajibTerminum ;
int mL, selisih, sisa;

int valButton;
int doorState;
int lastDoorState = LOW;

// MILLIS
#define INTERVAL_MESSAGE1 500
#define INTERVAL_MESSAGE2 1000
#define INTERVAL_REMINDER 15000 //15 detiik

unsigned long time_1 = 0;
unsigned long time_2 = 0;
unsigned long time_reminder = 0;
```

```
// Fungsi Setup
void setup() {
  Serial.begin(115200);
  lcd.init();
  lcd.backlight();

  // Setup WiFi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("WiFi connected");

  // Setup NTP
  timeClient.begin();
  timeClient.setTimeOffset(25200); // INDONESIA

  // Setup Pin
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BUTTON_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(DOOR_PIN, INPUT);

  notif();
}

void loop() {
  valButton = digitalRead(BUTTON_PIN);
  doorState = digitalRead(DOOR_PIN);
```

```
// Check door state
if (doorState != lastDoorState) {
  if (doorState == HIGH) {
    Serial.println("TERTUTUP");
  } else {
    Serial.println("TERBUKA");
    digitalWrite(BUZZER_PIN, LOW);
  }
  //lastDoorState = doorState;
}

// Matikan buzzer saat tombol ditekan
if (valButton == HIGH) {
  digitalWrite(BUZZER_PIN, LOW);
  Serial.println("Buzzer OFF: Button pressed");
}



if (millis() > time_1 + INTERVAL_MESSAGE1) {
  time_1 = millis();

  // Mengukur jarak menggunakan HC-SR04
  int jarak = measureDistance();
  if (jarak >= tinggiBotol) jarak = tinggiBotol;
  mL = map(jarak, 0, tinggiBotol, kebutuhan, 0);
  selisih = kebutuhan - mL; // jumlah air yang diminum
  sisa = kebutuhan - selisih; // sisa air dalam botol
 wajibTerminum = sisa;

  if (jarak > 20) {
```

```
  if (valButton == HIGH ) { // Ensure buzzer is only turned on if button is not pressed
    digitalWrite(BUZZER_PIN, LOW);
    Serial.println(" Buzzer OF");
  }
  else if (valButton == LOW && doorState == HIGH) { // Ensure buzzer is only turned on if button is not pressed
    digitalWrite(BUZZER_PIN, HIGH);
    Serial.println(" Buzzer ON");
  }
  if (doorState == LOW){ // Ensure buzzer is only turned on if button is not pressed
    digitalWrite(BUZZER_PIN, LOW);
    Serial.println(" Buzzer OFF");
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("  Botol Kosong   ");
  lcd.setCursor(0, 1);
  lcd.print("isi air minumnya");
  delay(2000);
} else if (jarak < 15) {
  digitalWrite(BUZZER_PIN, LOW);
}

Serial.print(Time);
Serial.println('\t');
Serial.print("jarak: ");
Serial.print(jarak);
Serial.print("cm ");
Serial.println('\t');
```

```
  Serial.print("Air terminum: ");
  Serial.print(selisih);
  Serial.print(" mL ");
  Serial.println('\t');
  Serial.print("perlu dikonsumsi: ");
  Serial.print(mL);
  Serial.print(" mL ");
  Serial.println('\t');
  Serial.print("button: ");
  Serial.print(valButton);
  Serial.print(" MAGNETIC DOOR: ");
  Serial.print(doorState);
  Serial.println("");
  delay(500);
}

if ((unsigned long)millis() - lcdSwitch >= periodLcdSwitch) {
  char buff[21];
  char vStr[9];
  char cStr[9];

  if (lcdToggle) {
    dtostrf(kebutuhan, 5, 2, vStr);
    dtostrf(selisih, 5, 2, cStr);

    sprintf(buff, "target : %s mL", vStr);
    lcdLine(1, buff);
    sprintf(buff, "terminum : %s mL", cStr);
    lcdLine(2, buff);
  } else {
    dtostrf(wajibTerminum, 5, 2, vStr);

    sprintf(buff, "Wajib: %s mL", vStr);
    lcdLine(1, buff);
    sprintf(buff, "   ", vStr);
    lcdLine(2, buff);
  }
  lcdToggle = !lcdToggle;
  lcdSwitch = millis();
}

if (millis() > time_2 + INTERVAL_MESSAGE2) {
  time_2 = millis();

  timeClient.update();
  Time = timeClient.getFormattedTime();
  int splitT = Time.indexOf(":");
  jamStamp = Time.substring(0, splitT);
  menitStamp = Time.substring(splitT + 1, Time.length() - 3);
  detikStamp = Time.substring(splitT + 4, Time.length());

  jam = jamStamp.toInt();
  menit = menitStamp.toInt();
  detik = detikStamp.toInt();

  if ( menit == 59 && detik >= 50 && detik <= 57 ) { // Pengecekan setiap 1 jam sekali
    lcd.clear();
    Serial.println("PENGINGAT INTERVAL 1 JAM");
    lcd.setCursor(0, 1);
    lcd.print("WAKTUNYA MINUM!!!");
```

```
      digitalWrite(BUZZER_PIN, HIGH);
      delay(10000);
      if (doorState == LOW){ // Ensure buzzer is only turned on if button is not pressed
        digitalWrite(BUZZER_PIN, LOW);
        Serial.println(" Buzzer OFF");
      }

    }
  }
  //Pengecekan interval detik pengingat minum
//   if(millis()> time_reminder + INTERVAL_REMINDER){
//    time_reminder = millis();
//    lcd.clear();
//    Serial.println("WAKTUNYA MINUM");
//     lcd.setCursor(0, 1);
//     lcd.print("WAKTUNYA MINUM!!!");
//    digitalWrite(BUZZER_PIN, HIGH);
//    delay(2000);
//   }
}

int measureDistance() {
  long duration, distance;
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  duration = pulseIn(ECHO_PIN, HIGH);
  distance = duration * 0.034 / 2;  // Konversi ke sentimeter
  return distance;
}

void lcdLine(uint8_t line, String text) {
  lcd.setCursor(0, line - 1);
  lcd.print("                ");
  lcd.setCursor(0, line - 1);
  lcd.print(text);
}

void notif() {
  digitalWrite(BUZZER_PIN, HIGH);
  Serial.println("Buzzer ON: Notification");
  delay(70);
  digitalWrite(BUZZER_PIN, LOW);
  Serial.println("Buzzer OFF: Notification");
  delay(50);
  digitalWrite(BUZZER_PIN, HIGH);
  Serial.println("Buzzer ON: Notification");
  delay(70);
  digitalWrite(BUZZER_PIN, LOW);
  Serial.println("Buzzer OFF: Notification");
}
```