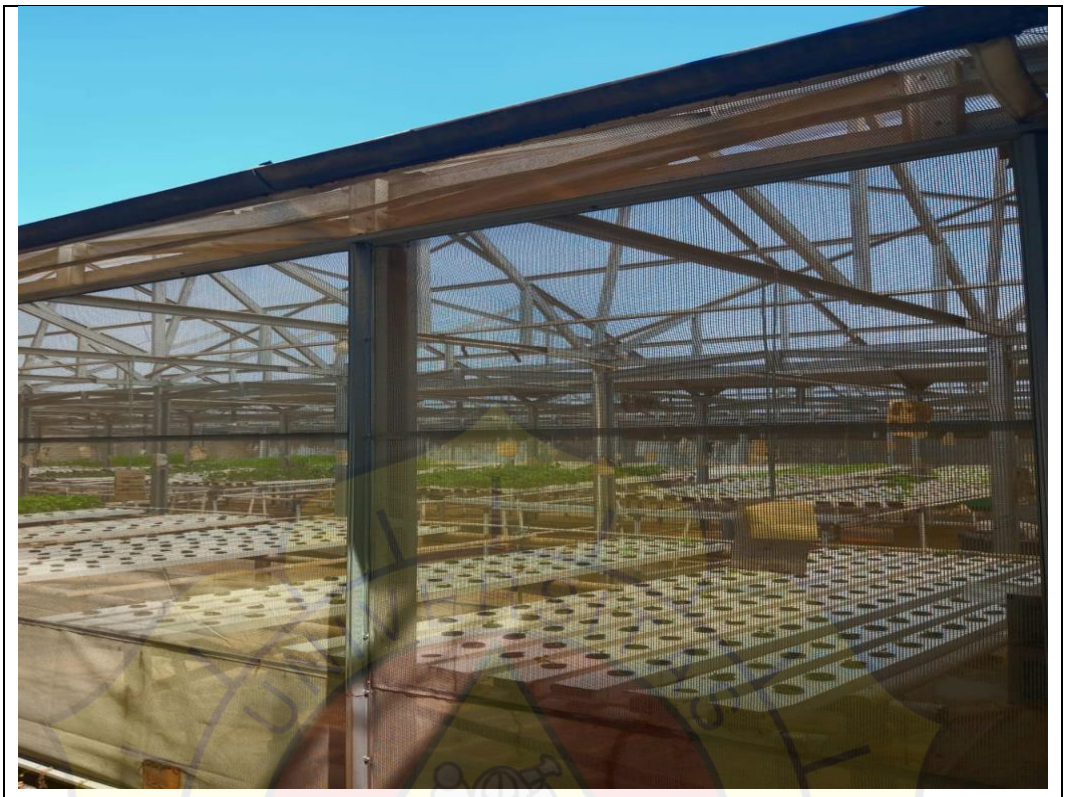
Lampiran 3 Tempat Studi kasus



Lampiran 4 Source Code

```
#include <Wire.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <DHT_U.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <EEPROM.h>
#include "GravityTDS.h"

// Konfigurasi WiFi
char ssid[] = "Ezerdam";
char pass[] = "ezerdam4389";

// Konfigurasi Blynk
char auth[] = "x7wvlS6-87GefvZ7biKz6fG00Oji7Y7v";

// Konfigurasi Pin
#define DHT_PIN 23  // Pin yang terhubung ke sensor DHT22
#define DS18B20_PIN 15  // Pin DS18B20 terhubung ke pin D15
#define TDS_SENSOR_PIN 35  // Pin D35 untuk sensor TDS Gravity
#define EEPROM_SIZE 512 //kalibrasi
#define SENSOR_POWER 3.3  // Pin yang terhubung ke power sensor
DS18B20
#define DHT_TYPE DHT22  // Tipe sensor DHT 22

#define RELAY_PIN_1 2
#define RELAY_PIN_2 4
#define RELAY_PIN_3 5
#define RELAY_PIN_4 18
#define RELAY_PIN_5 13
#define BUZZER_PIN 12

GravityTDS gravityTds;
DHT_Unified dht(DHT_PIN, DHT_TYPE);
OneWire oneWire(DS18B20_PIN);
DallasTemperature sensors(&oneWire);

//Potential Hydrogen
```

```
const int ph_Pin = 34;
float Po = 0; //pH Output
float PH_step;
int nilai_analog_Ph;
double TeganganPH;

float temperature = 25, tdsValue = 0;
float PH4 = 3.299;
float PH7 = 2.7;
bool buzzerState = false;

void setup() {
  Serial.begin(115200);
  EEPROM.begin(EEPROM_SIZE);  //Initialize EEPROM
  // Inisialisasi sensor DHT22
  dht.begin();
  // Koneksi ke WiFi
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(250);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");
  // Koneksi ke Blynk
  Blynk.begin(auth, ssid, pass);

  // Inisialisasi pin power sensor DS18B20
  pinMode(SENSOR_POWER, OUTPUT);
  digitalWrite(SENSOR_POWER, HIGH);
  sensors.begin();

  // Inisialisasi sensor TDS
  gravityTds.setPin(TDS_SENSOR_PIN);
  gravityTds.setAref(3.3);
  gravityTds.setAdcRange(4096);
  gravityTds.begin();

  //PH
  pinMode(ph_Pin, INPUT);
```

```
  // Inisialisasi pin relay sebagai OUTPUT
  pinMode(RELAY_PIN_1, OUTPUT);
  pinMode(RELAY_PIN_2, OUTPUT);
  pinMode(RELAY_PIN_3, OUTPUT);
  pinMode(RELAY_PIN_4, OUTPUT);
  pinMode(RELAY_PIN_5, OUTPUT);

  // Inisialisasi pin buzzer sebagai OUTPUT
  pinMode(BUZZER_PIN, OUTPUT);

  // Awalnya, matikan semua relay
  digitalWrite(RELAY_PIN_1, HIGH);
  digitalWrite(RELAY_PIN_2, HIGH);
  digitalWrite(RELAY_PIN_3, HIGH);
  digitalWrite(RELAY_PIN_4, HIGH);
  digitalWrite(RELAY_PIN_5, LOW);

  // Matikan buzzer
  digitalWrite(BUZZER_PIN, LOW);
}

void loop() {
  Blynk.run();
  if (buzzerState) {
    playBuzzerPattern();
  }
  // Membaca data dari sensor DHT22
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  float temperatureDHT = event.temperature;
  dht.humidity().getEvent(&event);
  float humidityDHT = event.relative_humidity;

  // Membaca data dari sensor DS18B20
  sensors.requestTemperatures();
  float temperatureDS18B20 = sensors.getTempCByIndex(0);

  // Membaca nilai TDS dari sensor Gravity TDS
```

```
  gravityTds.setTemperature(temperature);
  gravityTds.update();
  tdsValue = gravityTds.getTdsValue();

  //Membaca nilai PH
  nilai_analog_Ph = analogRead(ph_Pin);
  TeganganPH = 3.3 / 4096 * nilai_analog_Ph;
  PH_step = (PH4 - PH7) / 3;
  Po = 7.00 + ((PH7 - TeganganPH) / PH_step);

  // Menampilkan data pada Serial Monitor
  Serial.print("DHT Temperature: ");
  Serial.print(temperatureDHT);
  Serial.println(" °C");
  Serial.print("DHT Humidity: ");
  Serial.print(humidityDHT);
  Serial.println(" %");
  Serial.print("DS18B20 Temperature: ");
  Serial.print(temperatureDS18B20);
  Serial.println(" °C");
  Serial.print("TDS Value: ");
  Serial.print(tdsValue, 0);
  Serial.println(" ppm");
  Serial.print("Nilai PH: ");
  Serial.println(Po, 2);

  // Mengirim data ke Blynk
  Blynk.virtualWrite(V0, temperatureDHT);
  Blynk.virtualWrite(V1, humidityDHT);
  Blynk.virtualWrite(V2, temperatureDS18B20);
  Blynk.virtualWrite(V3, tdsValue);
  Blynk.virtualWrite(V4, Po);
  // Menunggu sejenak sebelum membaca ulang sensor
  delay(3000);
}
BLYNK_WRITE(V5) {
  int relayState = param.asInt();
  digitalWrite(RELAY_PIN_1, relayState);
}
```

```
BLYNK_WRITE(V6) {
  int relayState = param.asInt();
  digitalWrite(RELAY_PIN_2, relayState);
}
BLYNK_WRITE(V7) {
  int relayState = param.asInt();
  digitalWrite(RELAY_PIN_3, relayState);
}
BLYNK_WRITE(V8) {
  int relayState = param.asInt();
  digitalWrite(RELAY_PIN_4, relayState);
}
BLYNK_WRITE(V10) {
  int relayState = param.asInt();
  digitalWrite(RELAY_PIN_5, relayState);
}
BLYNK_WRITE(V9) {
  buzzerState = param.asInt();
}
void playBuzzerPattern() {
  for (int i = 0; i < 3; i++) {
    digitalWrite(BUZZER_PIN, HIGH);
    delay(100);
    digitalWrite(BUZZER_PIN, LOW);
    delay(100);
  }
  delay(500);
}
```