**LAPORAN JARINGAN SENSOR NIRKABEL**

**PROTOTYPE LAMPU CERDAS**



**OLEH:**

Muh. Khaeril Syam (D42116008)

Lutfi Qadri (D42116012)

Andi Muh. Agung Alif H. (D42116502)

Dandi Wisnu Shoreandi (D42116512)

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS HASANUDDIN**

**2019**

1. Alat dan Bahan
2. NodeMCU V3 (2)
3. Sensor Ultrasonic
4. Relay Module 1 channel
5. Kabel / Jumper
6. Kabel USB (2)
7. Lampu
8. Aplikasi Blynk
9. Langkah-langkah Pengerjaan
10. Rangkaikan NodeMCU, Relay dan lampu menggunakan kabel/jumper.

**NodeMCU Relay Lampu**

Pin 3V VCC -

Pin D4 Pin IN -

Pin GND GND -

- NO Anoda Lampu

- COM Sumber tegangan

1. Rangkaian NodeMCU dan sensor ultrasonic menggunakan kabel/jumper

**NodeMCU Sensor Ultrasonic**

Pin D3 Pin Echo

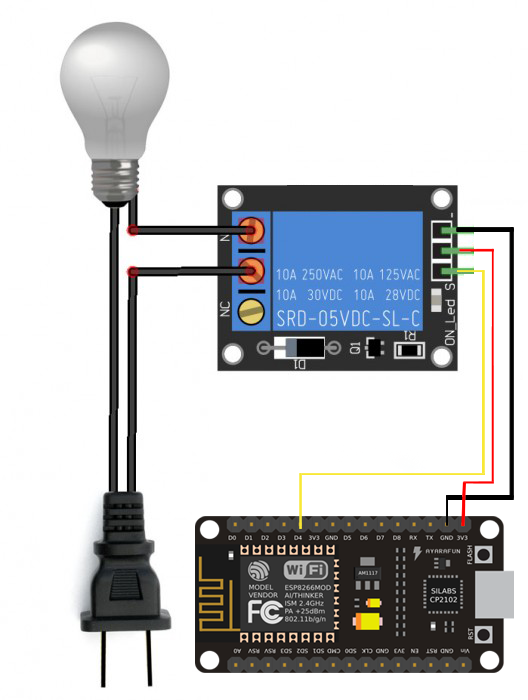
Pin D4 Pin Trig

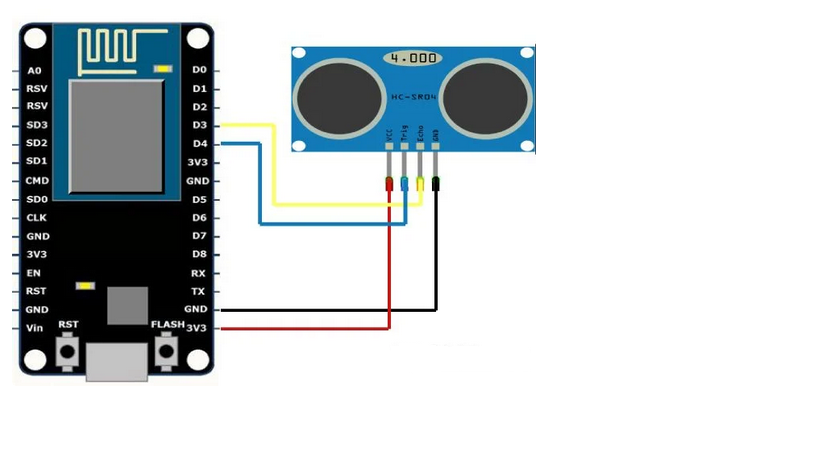
Pin 3V Pin VCC

Pin GND Pin GND

1. Selanjutnya Upload masing-masing sketch program ke NodeMCU
2. Selanjutnya rangkaiakan masing-masing alat dan bahan yang sesuai dengan masing-masing letak fungsinya.
3. Atur Pin, Acces Point dan token pada aplikasi Blynk.
4. Cara kerja alat

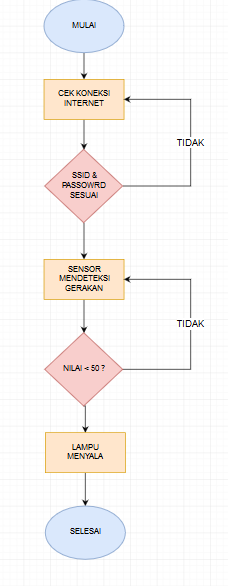
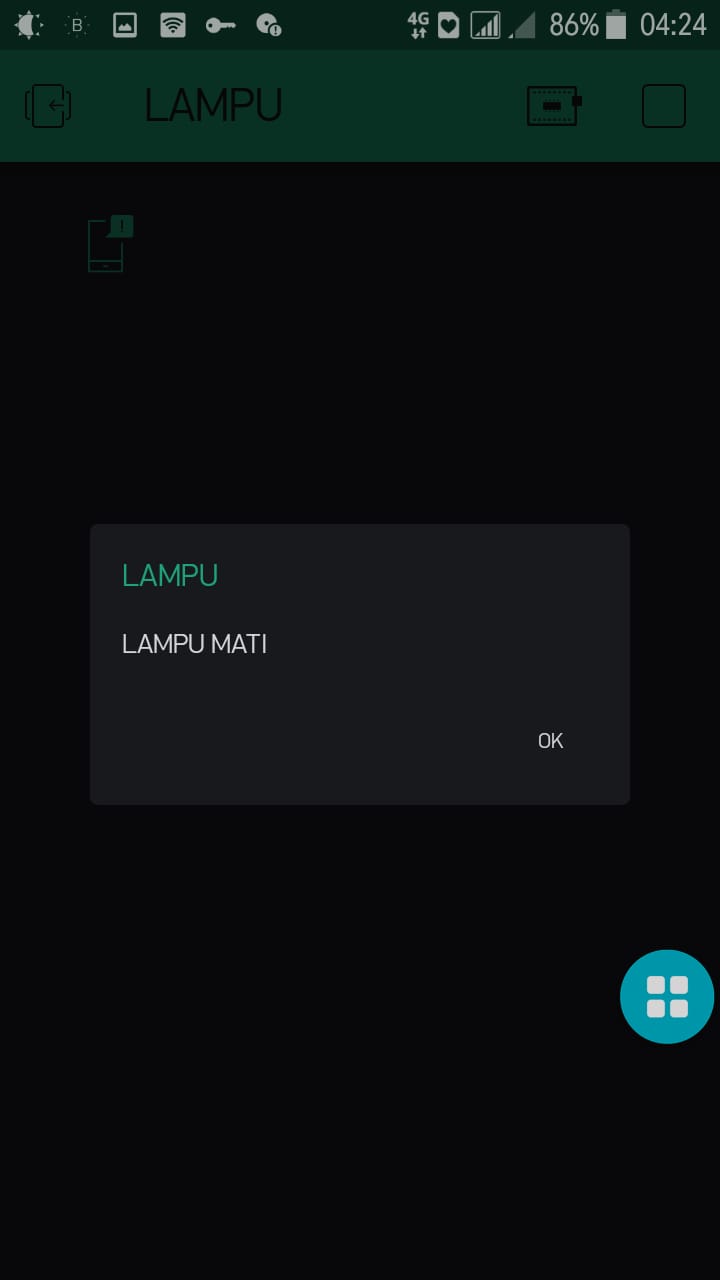
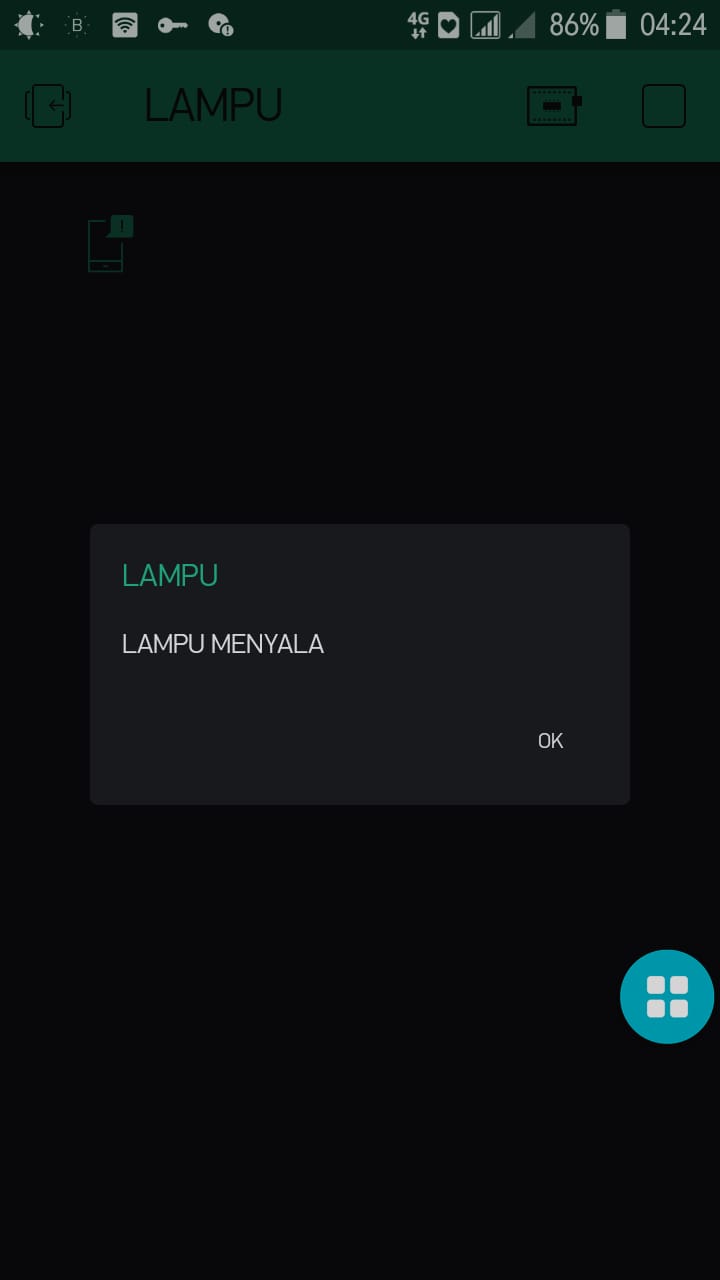
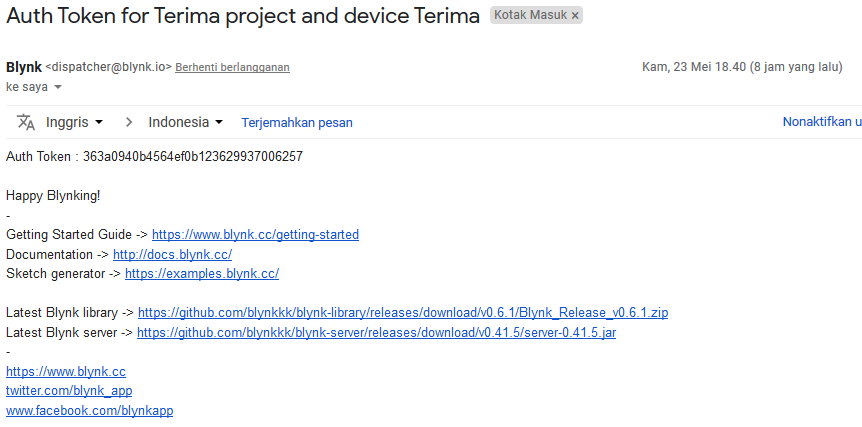
Alat ini bekerja sebagai otomatisasi lampu, dengan menggunakan sensor ultrasonic sebagai sensor utama dan menggunakan dua boar mikrokontroller jenis NodeMCU, dimana salah satu NodeMCU terhubung dengan sensor dan NodeMCU yang lain terhubung dengan relay yang akan mengaktifkan lampu, jadi jika sensor ultrasonic mendeteksi pergerakan maka data akan terkirim ke NodeMCU dan mengaktifkan lampu.



1. Skema rangkaian
2. NodeMCU, relay dan lampu
3. NodeMCU dan sensor ultrasonic
4. Hasil
5. NodeMCU, relay dan lampu

Lampu mati

Lampu menyala

1. NodeMCU, sensor ultrasonic
2. Flowchart
3. Tampilan pada aplikasi Blynk
4. Token di Surat elektronik
5. Token NodeMCU, sensor ultrasonic (sebagai server)
6. Token NodeMCU, Relay dan Lampu (sebagai client)
7. Source Program
8. NodeMCU, relay dan lampu (sebagai client)

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

char auth[] = "363a0940b4564ef0b123629937006257";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "acer";

char pass[] = "sayasaya";

bool touched = false;

void setup()

{

// Debug console

Serial.begin(9600);

pinMode(2,OUTPUT);

digitalWrite(2,00);

Blynk.begin(auth, ssid, pass);

}

void loop()

{

Blynk.run();

}

BLYNK\_WRITE(V1)

{

int pinValue = param.asInt(); // assigning incoming value from pin V1 to a variable

AturRelay(pinValue);

// process received value

}

void AturRelay(int pinValue){

if(pinValue == 1){

digitalWrite(2,1);

Blynk.notify("LAMPU MENYALA");

// digitalWrite(2,HIGH);

}//NYALA

else if(pinValue == 0){

digitalWrite(2,00);

Blynk.notify("LAMPU MATI");

// digitalWrite(2,LOW);

}//MATI

}

1. NodeMCU, Sensor Ultrasonic (sebagai server)

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.

// Go to the Project Settings (nut icon).

//DEFINE PINS

const int trigPin = 2; //D4

const int echoPin = 0; //D3

//DEFINE VARIABLES

long duration, distance;

char auth[] = "8332411289ea46ccae43f96799072c0f";

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "acer";

char pass[] = "sayasaya";

bool switches = false; // false = relay mati (HIGH) , true = relay nyala (LOW)

WidgetBridge bridge1(V1);

// Timer for blynking

BlynkTimer timer; BLYNK\_READ(V1)

{

// This command writes Arduino's uptime in seconds to Virtual Pin (5)

}

void setup()

{

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

// Debug console

Serial.begin(9600);

Blynk.begin(auth, ssid, pass); timer.setInterval(1000L, blynkAnotherDevice);

}

void loop()

{

Blynk.run(); timer.run();

}

void blynkAnotherDevice() // Here we will send HIGH or LOW once per second

{

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

distance = duration\*0.034/2;

Serial.println(distance);

if(distance > 50){ // jika nilai lebih dari 50 maka lampu mati

// Serial.println(SensorValue);

bridge1.digitalWrite(V1, HIGH); //

bridge1.virtualWrite(V1,1);

}else if(distance < 51 ){ // jika nilai lebih dari 50 maka lampu menyala

bridge1.digitalWrite(V1, LOW);

bridge1.virtualWrite(V1,0);

}

}

BLYNK\_CONNECTED() {

bridge1.setAuthToken("363a0940b4564ef0b123629937006257");

// Place the AuthToken of the second hardware here

}