Lampiran – lampiran

Code program alat

```
/*************************************************************
  Download latest Blynk library here:
    https://github.com/blynkkk/blynk-library/releases/latest

  Blynk is a platform with iOS and Android apps to control
  Arduino, Raspberry Pi and the likes over the Internet.
  You can easily build graphic interfaces for all your
  projects by simply dragging and dropping widgets.

    Downloads, docs, tutorials: http://www.blynk.cc
    Sketch generator:           http://examples.blynk.cc
    Blynk community:            http://community.blynk.cc
    Follow us:                  http://www.fb.com/blynkapp
                                http://twitter.com/blynk_app

  Blynk library is licensed under MIT license
  This example code is in public domain.

*************************************************************
smart sensor parkir menggunakan ultrasonic
created by N,HUDA for educational purpose

  Note: This requires ESP32 support package:
    https://github.com/espressif/arduino-esp32

*************************************************************/
```

```
/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial

/* Fill-in your Template ID (only if using Blynk.Cloud) */
#define BLYNK_TEMPLATE_ID "TMPLrBnr4N0E"
#define BLYNK_DEVICE_NAME "Parkir NHuda"

//deklarasi pin sensor ultrasonic
#define trigPin1 4
#define echoPin1 16
#define trigPin2 17
#define echoPin2 18
#define trigPin3 19
#define echoPin3 21
#define trigPin4 22
#define echoPin4 23
//deklarasi pin LED
#define LED1 12
#define LED2 14
#define LED3 27
#define LED4 26

//kecepatan cahaya dalam cm/uS
#define SOUND_SPEED 0.034

//variabel saat membaca sensor
long duration, sensor1, sensor2, sensor3, sensor4;
float distanceCm;



#include <WiFi.h>
```

```
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "N5FLJ1mtsuO8OGccODZqjzDMzZ45oUox";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "arduino";
char pass[] = "ESAUNGGUL";

void setup()
{
  // Debug console
  Serial.begin(9600);
  delay(100);
  //deklarasi pin trigger sebagai output dan pin echo sebagai input
  pinMode(trigPin1, OUTPUT); //
  pinMode(echoPin1, INPUT); //
  pinMode(trigPin2, OUTPUT); //
  pinMode(echoPin2, INPUT); //
  pinMode(trigPin3, OUTPUT); //
  pinMode(echoPin3, INPUT); //
  pinMode(trigPin4, OUTPUT); //
  pinMode(echoPin4, INPUT); //

  //DEKLARASI PIN LED sebagai pin OUTPUT
  pinMode(LED1, OUTPUT); //
  pinMode(LED2, OUTPUT); //
  pinMode(LED3, OUTPUT); //
```

```
 pinMode(LED4, OUTPUT); //


 Blynk.begin(auth, ssid, pass);
}

void loop()
{
   // nilai pada sensor 1-4
 SonarSensor(trigPin1, echoPin1);
 sensor1=distanceCm;
 SonarSensor(trigPin2, echoPin2);
 sensor2=distanceCm;
 SonarSensor(trigPin3, echoPin3);
 sensor3=distanceCm;
 SonarSensor(trigPin4, echoPin4);
 sensor4=distanceCm;


 if(sensor1<=20){      //jika ada kendataan terdeteksi di sensor 1
   Blynk.virtualWrite(V0, 1);
   digitalWrite(LED1, HIGH);
 }else if(sensor1>20){ //jika tidak ada kendataan terdeteksi di sensor 1
   Blynk.virtualWrite(V0, 0);
   digitalWrite(LED1, LOW);
 }

 if(sensor2<=20){      //jika ada kendataan terdeteksi di sensor 2
   Blynk.virtualWrite(V1, 1);
   digitalWrite(LED2, HIGH);
 }else if(sensor2>20){ //jika tidak ada kendataan terdeteksi di sensor 2
```

```
  Blynk.virtualWrite(V1, 0);
  digitalWrite(LED2, LOW);
 }

 if(sensor3<=20){      //jika ada kendataan terdeteksi di sensor 3
  Blynk.virtualWrite(V2, 1);
  digitalWrite(LED3, HIGH);
 }else if(sensor3>20){ //jika tidak ada kendataan terdeteksi di sensor 3
  Blynk.virtualWrite(V2, 0);
  digitalWrite(LED3, LOW);
 }

 if(sensor4<=20){      //jika ada kendataan terdeteksi di sensor 4
  Blynk.virtualWrite(V3, 1);
  digitalWrite(LED4, HIGH);
 }else if(sensor4>20){ //jika tidak ada kendataan terdeteksi di sensor 4
  Blynk.virtualWrite(V3, 0);
  digitalWrite(LED4, LOW);
 }
 Blynk.run();
}

//fungsi untuk membaca nilai sensor ultrasonic
void SonarSensor(int trigPin, int echoPin){
 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);

  // Calculate the distance
  distanceCm = duration * SOUND_SPEED/2;
}
```