# LAMPIRAN 1

# CODING

```
#include "esp_camera.h"
#include <WiFi.h>
#include "FirebaseESP32.h"

//
// WARNING!!! Make sure that you have either selected ESP32 Wrover Module,
//            or another board which has PSRAM enabled
//

// Select camera model
//#define CAMERA_MODEL_WROVER_KIT
//#define CAMERA_MODEL_ESP_EYE
//#define CAMERA_MODEL_M5STACK_PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE
#define CAMERA_MODEL_AI_THINKER

#include "camera_pins.h"

#define FIREBASE_HOST "motionalarm-9d113.firebaseio.com"
#define FIREBASE_AUTH "kFnJuAOJsKsJeNk0C4MNNvADuaQv8KUefAJWHxEP"

#define pir 15   // Data Pin of PIR 15 , for ESP32CAM GPIO no. is 15

const char* ssid = "skripsi";
const char* password = "skripsi123";

void startCameraServer();
```

```
FirebaseData firebaseData;

int stateMotion = LOW;             // default tidak ada gerakan
int valMotion = 0;

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
  //init with high specs to pre-allocate larger buffers
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
```

```
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

pinMode(pir,INPUT);

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  sensor_t * s = esp_camera_sensor_get();
  //initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1);//flip it back
    s->set_brightness(s, 1);//up the blightness just a bit
    s->set_saturation(s, -2);//lower the saturation
  }
  //drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE)
  s->set_vflip(s, 1);
```

```
  s->set_hmirror(s, 1);
#endif

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);

  //Set database read timeout to 1 minute (max 15 minutes)
  Firebase.setReadTimeout(firebaseData, 1000 * 60);
  //tiny, small, medium, large and unlimited.
  //Size and its write timeout e.g. tiny (1s), small (10s), medium (30s)
and large (60s).
  Firebase.setwriteSizeLimit(firebaseData, "tiny");
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(500);

  valMotion = digitalRead(pir);   // read sensor value
```

```
  if (valMotion == HIGH) {           // check if the sensor is HIGH
    if (stateMotion == LOW) {
      Firebase.setInt(firebaseData, "/pir", 1);
      //man=1; Firebase.setFloat ("/GUN/pir",man);
      stateMotion = HIGH;       // update variable state to HIGH
    }
  }
  else {
    if (stateMotion == HIGH){
      Firebase.setInt(firebaseData, "/pir", 0);
      //man=0; Firebase.setFloat ("/GUN/pir",man);
      stateMotion = LOW;       // update variable state to LOW
    }
  }
}
```