

Exclude quotes Off  
Exclude bibliography Off

Exclude matches Off

### Lampiran 3 Source Code ESP32

```
1 | #include <ESP32Servo.h>
2 | #include <WiFi.h>
3 | #include <WiFiClient.h>
4 | #include <DHT.h>
5 | #include "HX711.h"
6 | // konfigurasi blynk & wifi
7 | #define BLYNK_TEMPLATE_ID "TMPL6gl8mcQii"
8 | #define BLYNK_TEMPLATE_NAME "Automatic Cat Feeder"
9 | #define BLYNK_AUTH_TOKEN
   "bKPlpxwPiJGaPbrccv50znnUrRYMLy"
10 | #include <BlynkSimpleEsp32.h>
11 | char ssid[] = "Jasmine";
12 | char pass[] = "nicolezef";
13 | // konfigurasi servo pakan
14 | Servo myservoPakan;
15 | int StatusPakan;
16 | // konfigurasi dht22
17 | #define DHTPIN 15
18 | #define DHTTYPE DHT22
19 | DHT dht(DHTPIN, DHTTYPE);
20 | // konfigurasi ultrasonik
21 | const int trigPin = 33;
22 | const int echoPin = 32;
23 | long duration;
24 | float distance;
25 | long sisaPakan;
26 | long tinggiSensor = 11;
27 | // konfigurasi pir + servopir
28 | const int pirPin = 14;
29 | const int servoPinPIR = 12;
30 | Servo myServoPIR;
```

```

31 |     int pirState = LOW;
32 |     int val = 0;

33 |     // konfigurasi loadcell
34 |     const int doutPin = 4;
35 |     const int sckPin = 2;
36 |     HX711 scale;
37 |     float calibration_factor = 412.33;

38 |     void setup() {
39 |     Serial.begin(115200);

40 |     // Inisialisasi WiFi
41 |     WiFi.begin(ssid, pass);
42 |     while (WiFi.status() != WL_CONNECTED) {
43 |     Serial.print(".");
44 |     delay(500);
45 |     }
46 |     Serial.println("WiFi Terkoneksi");

47 |     // Inisialisasi Blynk
48 |     Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
49 |     Serial.println("Blynk Terkoneksi");

50 |     // Inisialisasi servo pemberian pakan
51 |     myservoPakan.attach(13);
52 |     myservoPakan.write(0);

53 |     // Inisialisasi sensor DHT
54 |     dht.begin();

55 |     // Inisialisasi sensor ultrasonik
56 |     pinMode(trigPin, OUTPUT);
57 |     pinMode(echoPin, INPUT);

58 |     // Inisialisasi sensor PIR dan servo terkait
59 |     pinMode(pirPin, INPUT);
60 |     myServoPIR.attach(servoPinPIR);
61 |     myServoPIR.write(0);

62 |     // Inisialisasi sensor LoadCell dan Modul HX711
63 |     scale.begin(doutPin, sckPin);
64 |     scale.set_scale(calibration_factor);
65 |     scale.tare();
66 |     }

67 |     void bukaWadah() {
68 |     for (int posisi = 0; posisi <= 95; posisi++) {
69 |     myServoPIR.write(posisi);
70 |     delay(20);
71 |     }
72 |     }

73 |     void tutupWadah() {

```

```

74 |     for (int posisi = 95; posisi >= 0; posisi--) {
75 |         myServoPIR.write(posisi);
76 |         delay(20);
77 |     }
78 | }

79 | void beriPakan() {
80 |     bukaWadah();
81 |     delay(500);
82 |     for(int posisi = 0; posisi <= 70; posisi++) {
83 |         myservoPakan.write(posisi);
84 |         delay(15);
85 |     }
86 |     // delay(1000);
87 |     for(int posisi = 70; posisi >= 0; posisi--) {
88 |         myservoPakan.write(posisi);
89 |         delay(15);
90 |     }
91 |     delay(500);
92 |     tutupWadah();
93 |     delay(1000);
94 | }

95 | void loop() {
96 |     Blynk.run();

97 |     // Sensor DHT
98 |     float h = dht.readHumidity();
99 |     float t = dht.readTemperature();
100 |     if (!isnan(h) && !isnan(t)) {
101 |         Serial.print("Humidity: ");
102 |         Serial.print(h);
103 |         Serial.print(" %\t");
104 |         Serial.print("Temperature: ");
105 |         Serial.print(t);
106 |         Serial.println(" *C");
107 |         Blynk.virtualWrite(V1, t);
108 |         Blynk.virtualWrite(V2, h);
109 |         if ((h >= 0 && h <= 50) || (h >= 65 && h <= 100)) {
110 |             Blynk.logEvent("lembab", "Kelembaban di kandang
berada di luar batas normal!");
111 |         }
112 |         if (t >= 0 && t <= 21) {
113 |             Blynk.logEvent("suhu_rendah", "Suhu di kandang
terlalu rendah!");
114 |         } else if (t >= 30 && t <= 35) {
115 |             Blynk.logEvent("suhu_tinggi", "Suhu di kandang
terlalu tinggi!");
116 |         }
117 |         } else {
118 |             Serial.println("Failed to read from DHT sensor!");
119 |         }

120 |     // Sensor ultrasonik

```

```

121 |     digitalWrite(trigPin, LOW);
122 |     delayMicroseconds(2);
123 |     digitalWrite(trigPin, HIGH);
124 |     delayMicroseconds(10);
125 |     digitalWrite(trigPin, LOW);
126 |     duration = pulseIn(echoPin, HIGH);
127 |     distance = duration * 0.034 / 2;
128 |     sisaPakan = tinggiSensor - distance;
129 |     Blynk.virtualWrite(V3, sisaPakan);
130 |     if (sisaPakan < 3) {
131 |         Serial.print("Sisa Pakan: ");
132 |         Serial.println(sisaPakan);
133 |         Blynk.logEvent("stok_kesediaan_pakan", "Sisa pakan
sudah dibawah batas, mohon diisi kembali");
134 |     } else {
135 |         Serial.print("Sisa Pakan: ");
136 |         Serial.println(sisaPakan);
137 |     }

138 |     // Sensor PIR
139 |     val = digitalRead(pirPin); // Membaca nilai dari
sensor PIR
140 |     if (val == HIGH) {
141 |         if (pirState == LOW) {
142 |             Serial.println("Gerakan terdeteksi!");
143 |             for (int posisi = 0; posisi <= 95; posisi++) {
144 |                 myServoPIR.write(posisi);
145 |                 delay(20);
146 |             }
147 |             pirState = HIGH;
148 |         }
149 |         delay(5000);
150 |     } else {
151 |         if (pirState == HIGH) {
152 |             Serial.println("Tidak ada gerakan.");
153 |             for (int posisi = 95; posisi >= 0; posisi--) {
154 |                 myServoPIR.write(posisi);
155 |                 delay(20);
156 |             }
157 |             pirState = LOW;
158 |         }
159 |     }
160 |     delay(500); // Penundaan untuk menghindari pembacaan
berlebihan

161 |     // Sensor LoadCell dan Modul HX711
162 |     float weight = scale.get_units(10); // Baca 10 kali
untuk hasil yang lebih stabil
163 |     Blynk.virtualWrite(V4, weight);
164 |     if (weight >18 && weight <20) {
165 |         Serial.print("Berat Pakan pada Wadah: ");
166 |         Serial.println(weight);
167 |         Blynk.logEvent("berat_pakan", "Berat wadah pakan
dibawah 20 gram, mohon dicek");

```

```

168 |     } else {
169 |         Serial.print("Berat Pakan pada Wadah: ");
170 |         Serial.println(weight);
171 |     }

172 |     delay(100);

173 |     // Pemberian pakan otomatis
174 |     if (StatusPakan == 1) {
175 |         beriPakan();
176 |         Blynk.logEvent("status_pemberian_pakan", "Kucing
telah diberi pakan");
177 |         Blynk.virtualWrite(V0, 0);
178 |         StatusPakan = 0;
179 |         delay(2000);
180 |     }
181 | }

182 | BLYNK_WRITE(V0) {
183 |     StatusPakan = param.asInt();
184 | }

```

#### Lampiran 4 Source Code ESP32Cam

```

1 |     #define BLYNK_TEMPLATE_ID "TMPL6gl8mcQii"
2 |     #define BLYNK_TEMPLATE_NAME "Automatic Cat Feeder"
3 |     #define BLYNK_AUTH_TOKEN "E-
dW2fC68NsvdlWVjq0uStWjOVsO4yue"
4 |     #define BLYNK_PRINT Serial

5 |     #include "src/OV2640.h"
6 |     #include <WiFi.h>
7 |     #include <WebServer.h>
8 |     #include <WiFiClient.h>
9 |     #include <BlynkSimpleEsp32.h>

10 |     #define CAMERA_MODEL_AI_THINKER

11 |     #include "camera_pins.h"

12 |     char auth[] = BLYNK_AUTH_TOKEN;
13 |     #define SSID1 "Jasmine"
14 |     #define PWD1 "nicolezef"

15 |     OV2640 cam;

16 |     WebServer server(80);

17 |     const char HEADER[] = "HTTP/1.1 200 OK\r\n" \
18 |     "Access-Control-Allow-Origin: *\r\n" \

```

```

19 |     "Content-Type:          multipart/x-mixed-replace;
    boundary=1234567890000000000000987654321\r\n";
20 |     const      char      BOUNDARY[]      =      "\r\n--
    12345678900000000000000987654321\r\n";
21 |     const      char      CTNTTYPE[]      =      "Content-Type:
    image/jpeg\r\nContent-Length: ";
22 |     const int  hdrLen = strlen(HEADER);
23 |     const int  bdrLen = strlen(BOUNDARY);
24 |     const int  cntLen = strlen(CTNTTYPE);

25 |     void handle_jpg_stream(void)
26 |     {
27 |     char buf[32];
28 |     int s;

29 |     WiFiClient client = server.client();

30 |     client.write(HEADER, hdrLen);
31 |     client.write(BOUNDARY, bdrLen);

32 |     while (true)
33 |     {
34 |     if (!client.connected()) break;
35 |     cam.run();
36 |     s = cam.getSize();
37 |     client.write(CTNTTYPE, cntLen);
38 |     sprintf( buf, "%d\r\n\r\n", s );
39 |     client.write(buf, strlen(buf));
40 |     client.write((char *)cam.getfb(), s);
41 |     client.write(BOUNDARY, bdrLen);
42 |     }
43 |     }

44 |     const char JHEADER[] = "HTTP/1.1 200 OK\r\n" \
45 |     "Content-disposition:          inline;
    filename=capture.jpg\r\n" \
46 |     "Content-type: image/jpeg\r\n\r\n";
47 |     const int  jhdLen = strlen(JHEADER);

48 |     void handle_jpg(void)
49 |     {
50 |     WiFiClient client = server.client();

51 |     cam.run();
52 |     if (!client.connected()) return;

53 |     client.write(JHEADER, jhdLen);
54 |     client.write((char *)cam.getfb(), cam.getSize());
55 |     }

56 |     void handleNotFound()
57 |     {
58 |     String message = "Server is running!\n\n";
59 |     message += "URI: ";

```

```

60 |     message += server.uri();
61 |     message += "\nMethod: ";
62 |     message += (server.method() == HTTP_GET) ? "GET" :
    "POST";
63 |     message += "\nArguments: ";
64 |     message += server.args();
65 |     message += "\n";
66 |     server.send(200, "text / plain", message);
67 | }

68 | void setup()
69 | {

70 |     Serial.begin(115200);
71 |     //while (!Serial);           //wait for serial
    connection.

72 |     camera_config_t config;
73 |     config.ledc_channel = LEDC_CHANNEL_0;
74 |     config.ledc_timer = LEDC_TIMER_0;
75 |     config.pin_d0 = Y2_GPIO_NUM;
76 |     config.pin_d1 = Y3_GPIO_NUM;
77 |     config.pin_d2 = Y4_GPIO_NUM;
78 |     config.pin_d3 = Y5_GPIO_NUM;
79 |     config.pin_d4 = Y6_GPIO_NUM;
80 |     config.pin_d5 = Y7_GPIO_NUM;
81 |     config.pin_d6 = Y8_GPIO_NUM;
82 |     config.pin_d7 = Y9_GPIO_NUM;
83 |     config.pin_xclk = XCLK_GPIO_NUM;
84 |     config.pin_pclk = PCLK_GPIO_NUM;
85 |     config.pin_vsync = VSYNC_GPIO_NUM;
86 |     config.pin_href = HREF_GPIO_NUM;
87 |     config.pin_sscb_sda = SIOD_GPIO_NUM;
88 |     config.pin_sscb_scl = SIOC_GPIO_NUM;
89 |     config.pin_pwdn = PWDN_GPIO_NUM;
90 |     config.pin_reset = RESET_GPIO_NUM;
91 |     config.xclk_freq_hz = 20000000;
92 |     config.pixel_format = PIXFORMAT_JPEG;

93 |     /*
94 |     if(psramFound()){
95 |     config.frame_size = FRAMESIZE_UXGA;
96 |     config.jpeg_quality = 10;
97 |     config.fb_count = 2;
98 |     } else {
99 |     config.frame_size = FRAMESIZE_SVGA;
100 |     config.jpeg_quality = 12;
101 |     config.fb_count = 1;
102 |     }
103 |     */

104 |     // Frame parameters
105 |     // config.frame_size = FRAMESIZE_UXGA;

```

```

106 |     config.frame_size          =          FRAMESIZE_SVGA;
      //UXGA, SXGA, XGA, SVGA, VGA, CIF, QVGA, HQVGA, QQVGA
107 |     config.jpeg_quality = 12;
108 |     config.fb_count = 2;

109 |     #if defined(CAMERA_MODEL_ESP_EYE)
110 |     pinMode(13, INPUT_PULLUP);
111 |     pinMode(14, INPUT_PULLUP);
112 |     #endif

113 |     cam.init(config);

114 |     IPAddress ip;

115 |     WiFi.mode(WIFI_STA);
116 |     WiFi.begin(SSID1, PWD1);
117 |     while (WiFi.status() != WL_CONNECTED)
118 |     {
119 |         delay(500);
120 |         Serial.print(F("."));
121 |     }
122 |     ip = WiFi.localIP();
123 |     Serial.println(F("WiFi connected"));
124 |     Serial.println("");
125 |     Serial.println(ip);
126 |     Serial.print("Stream Link: http://");
127 |     Serial.print(ip);
128 |     Serial.println("/mjpeg/1");
129 |     server.on("/mjpeg/1", HTTP_GET, handle_jpg_stream);
130 |     server.on("/jpg", HTTP_GET, handle_jpg);
131 |     server.onNotFound(handleNotFound);
132 |     server.begin();
133 |     }

134 |     void loop()
135 |     {
136 |         server.handleClient();
137 |     }

```