

LAMPIRAN

1. Anggaran Biaya

No	Nama	Jumlah	Harga Satuan	Biaya
1.	Arduino Uno	1	Buah	Rp.180.000
2.	Sensor <i>Real Time Clock</i>	1	Buah	Rp.45.000
3.	Sensor TCS3200	1	Buah	Rp.50.000
4	Sensor <i>Load Cell</i> dan Modul HX711	1	Buah	Rp.80.000
5	LM2596	1	Buah	Rp.20.000
6	LCDi2c	1	Buah	Rp.45.000
7	Motor Servo	1	Buah	Rp.54.000
8	Motor Water Pump	1	Buah	Rp.25.000
9	NodeMCU	1	Buah	Rp.46.000
10	ESP 32 CAM	1	Buah	Rp.270.00
11	Buzzer	1	Buah	Rp.15.000
Total				Rp.830.000

2. Daftar Riwayat Hidup

NAMA LENGKAP : HANDAVA WARDANA
NIM : 061940351947
TEMPAT TANGGAL LAHIR : PALEMBANG, 23 JULI 2001
ALAMAT : JL. TEMBOK BARU
GG.BERSAMA RT.17 RW.04
KOTA PALEMBANG
NO.HP : 0895420110404

RIWAYAT PENDIDIKAN FORMAL

PENDIDIKAN	NAMA SEKOLAH	TAMAT TAHUN
SD	SD NEGERI 84 PALEMBANG	2013
SMP	SMP NEGERI 7 PALEMBANG	2016
SMA	SEKOLAH MENENGAH ATAS NEGERI 19 PALEMBANG	2019

PENGHARGAAN/PENCAPAIAN

No.	PENGHARGAAN	PENGHARGAAN
1.	LOMBA LTBB PRAMUKA PENEGAK	JUARA 1

PENGALAMAN ORGANISASI

No	NAMA ORGANISASI	TAHUN
1	PRASIMBELS SMAN 19 PALEMBANG	2016-2018
2	ANGGOTA ROBOTIK POLSRI	2021-2022

Semua data yang saya tulis dan isi riwayat hidup ini benar dan dapat dipertanggung jawabkan.

Palembang, Agustus 2023

(Handava Wardana)



[JITEKI] Editor Decision

Kotak Masuk



Alfian Ma'arif Kemarin

kepada Irma, saya, Ahmad ▾



Dear dava Handava dava Wardana,

We have reached a decision regarding your submission to Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, "Cat Feeding Using Microcontroller Arduino Uno TCS3200 Sensor and Internet of Things".

Our decision is to: Accept Submission.

Send docx file to email alfianmaarif@ee.uad.ac.id

Thank you.

Best regards,
Assist. Prof. Alfian Ma'arif
Universitas Ahmad Dahlan
alfian.maarif@te.uad.ac.id

UNIVERSITAS AHMAD DAHLAN

Kampus 1: Jln. Kapas No. 9 Yogyakarta

Kampus 2: Jl. Pramuka 42, Sidikan, Umbulharjo, Yogyakarta 55161

Kampus 3: Jl. Prof. Dr. Soepomo, S.H., Janturan, Warungboto, Umbulharjo, Yogyakarta 55164

Kampus 4: Jl. Ringroad Selatan, Yogyakarta

Kampus 5: Jl. Ki Ageng Pemanahan 19, Yogyakarta

Kontak

Email: info@uad.ac.id

Telp. : (0274) 563515, 511830, 379418, 371120

Fax. : (0274) 564604



Balas



Balas ke semua



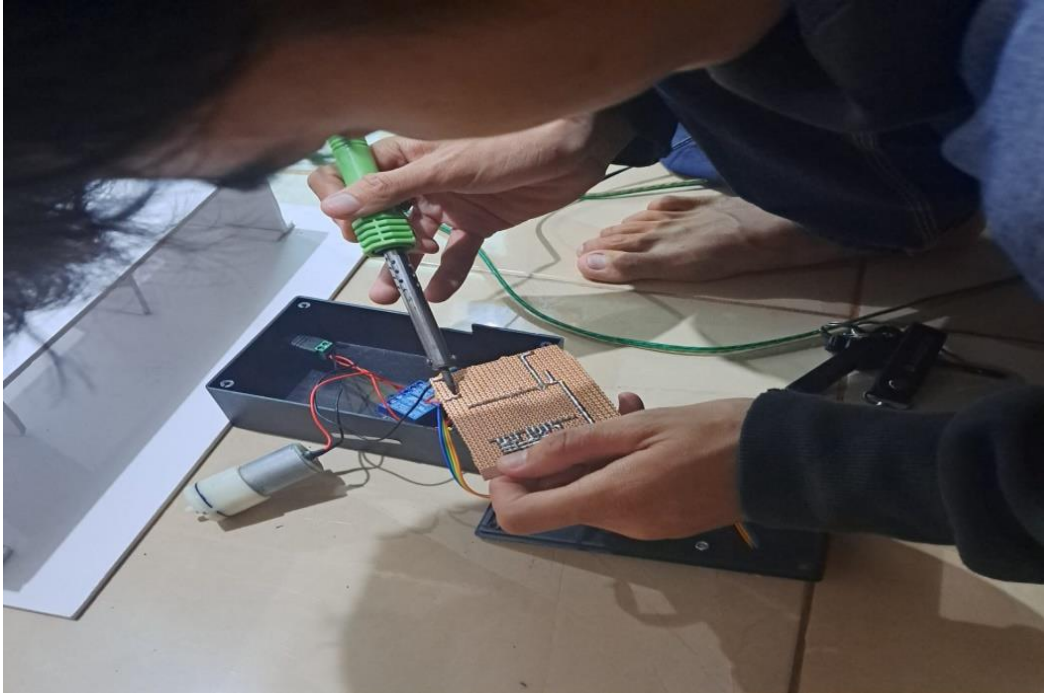
Teruskan



99+



PROSES PEMBUATAN RANGKAIAN



Survey Tempat Pet Shop



PROGRAM

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "HX711.h"
#include <DS3231.h> //mengincludekan library DS3231
#include <Servo.h>

Servo myservo;
DS3231 rtc(SDA, SCL); // inialisasi penggunaan i2c
int pakan = 0;
Time t;
int cam = 7;
int motor = 2;
int motorVcc = 3;
// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 20, 4);
HX711 scale(A1, A0);
float calibration_factor = 344.10;
int GRAM;
int pos;
int logicGram = 0;
//TCS
#define S0 8
#define S1 9
#define S2 10
#define S3 11
#define OutputSensor 12
#define Trig 4
#define Echo 5
long timer;
int jarak = 0;
int f_red = 0;
int f_green = 0;
int f_blue = 0;
int Jeda = 0;
#define Buzzer A2

void setup()
{
  // initialize the LCD
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
  pinMode(Buzzer, OUTPUT);
  lcd.init();
  scale.set_scale();
  // scale.tare();
  myservo.attach(6);

  // Turn on the backlight and print a message.
  lcd.backlight();
  lcd.print("Pakan Kucing");
  scale.set_scale(2280.f); // this value is obtained by calibrating the scale with known
weights; see the README for details
  // scale.tare();
  lcd.clear();
}
```

```

//TCS
Serial.begin(9600);
pinMode(S0, OUTPUT);
pinMode(S1, OUTPUT);
pinMode(S2, OUTPUT);
pinMode(S3, OUTPUT);
pinMode(OutputSensor, INPUT);
digitalWrite(S0, HIGH);
digitalWrite(S1, LOW);
pinMode(cam, OUTPUT);
digitalWrite(cam, HIGH);
pinMode(motor, OUTPUT);
pinMode(motorVcc, OUTPUT);
digitalWrite(Buzzer, LOW);
digitalWrite(motor, LOW);
digitalWrite(motorVcc, LOW);
//RTC
rtc.begin();
// setting pertama download program
//   rtc.setDate(8, 7, 2023); //mensetting tanggal 07 april 2018
//   rtc.setTime(20, 25, 00); //menset jam 22:00:00
//   rtc.setDOW(7); //menset hari "Sabtu"
//setelah didownload awal selesai, download kedua dengan memberi tanda komen "/"
Jeda = 0;
}

void loop()
{
  RTC1();
  loadcell();
  TCS();
  Jarak();
  digitalWrite(cam, HIGH);
  lcd.setCursor(0, 0); lcd.print(GRAM); lcd.print(" "); lcd.setCursor(5, 0); lcd.print(" g ");
  lcd.setCursor(0, 1); lcd.print(rtc.getDOWStr()); lcd.print(" ");
  lcd.setCursor(8, 1); lcd.print(rtc.getDateStr()); lcd.print(" ");
  lcd.setCursor(0, 2); lcd.print(rtc.getTimeStr()); lcd.print(" ");

  // 27 34 34
  if (f_red >= 23 && f_red <= 26 && f_green >= 20 && f_green <= 26 && f_blue >= 23 &&
f_blue <= 24) {
    lcd.setCursor(0, 3); lcd.print("Ada Kucing ");
    delay(3000);
  }
  if (f_red >= 112 && f_red <= 128 && f_green >= 175 && f_green <= 205 && f_blue >= 195
&& f_blue <= 225) {
    lcd.setCursor(0, 3); lcd.print("Ada Kucing ");
    delay(3000);
  }

  if (t.hour == 7 && t.min == 0 && t.sec == 0) {
    pakan = 1;
    lcd.setCursor(0, 3); lcd.print("Waktu Makan ");
  }
}

```

```

}
if (t.hour == 12 && t.min == 0 && t.sec == 0) {
  pakan = 1;
  lcd.setCursor(0, 3); lcd.print("Waktu Makan    ");
}
if (t.hour == 17 && t.min == 0 && t.sec == 0) {
  pakan = 1;
  lcd.setCursor(0, 3); lcd.print("Waktu Makan    ");
}
if (jarak <= 3) {
  pakan = 1;
  lcd.setCursor(0, 3); lcd.print("Ada Kucing    ");
}
if (GRAM < 200 && logicGram == 0) {
  digitalWrite(cam, LOW);
  digitalWrite(Buzzer, HIGH);
  delay(5000);
  logicGram = 1;
}
if (GRAM >= 200) {
  logicGram = 0;
  digitalWrite(Buzzer, LOW);
}
if (pakan == 1 && t.min >= Jeda) {
  Jeda = t.min + 2;
  if (Jeda >= 59) {
    Jeda = 2;
  }
  lcd.setCursor(0, 3); lcd.print("Menuangkan Makanan    ");
  myservo.write(180);
  delay(2000);
  myservo.write(0);
  delay(500);
  myservo.write(50);
  delay(1000);
  myservo.write(0);
  delay(1000);
  myservo.write(50);
  delay(1000);
  myservo.write(0);
  delay(1000);
  myservo.write(50);
  delay(1000);
  myservo.write(0);
  delay(1000);
  myservo.write(50);
  delay(1000);
  lcd.setCursor(0, 3); lcd.print("Makanan Siap    ");
  digitalWrite(motor, HIGH);
  digitalWrite(motorVcc, HIGH);
  delay(3000);
}

```



```

digitalWrite(motor, LOW);
digitalWrite(motorVcc, LOW);
lcd.setCursor(0, 3); lcd.print("          ");
pakan = 0;
}
if (pakan == 1 && t.min < Jeda) {
  lcd.setCursor(0, 3); lcd.print("Makanan Sudah Keluar");
  delay(3000);
  pakan = 0;
}
if (pakan == 0) {
  myservo.write(180);
  digitalWrite(motor, HIGH);
  lcd.setCursor(0, 3); lcd.print("          ");
}
if (t.min >= Jeda) {
  Jeda = 0;
}
}

void loadcell() {
  scale.set_scale(calibration_factor);
  GRAM = scale.get_units(), 4;
  GRAM = GRAM - 309;
  Serial.println(GRAM);
}

void TCS() {
  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);
  f_red = pulseIn(OutputSensor, LOW);
  Serial.print("R = ");
  Serial.println(f_red);
  delay(100);

  digitalWrite(S2, HIGH);
  digitalWrite(S3, HIGH);
  f_green = pulseIn(OutputSensor, LOW);
  Serial.print("G = ");
  Serial.println(f_green);
  delay(100);

  digitalWrite(S2, LOW);
  digitalWrite(S3, HIGH);
  f_blue = pulseIn(OutputSensor, LOW);
  Serial.print("B = ");
  Serial.println(f_blue);
  delay(100);
}

void RTC1() {
  Serial.print(rtc.getDOWStr()); //prosedur pembacaan hari
  Serial.print(" ");

  Serial.print(rtc.getDateStr()); //prosedur pembacaan tanggal
  Serial.print(" -- ");

```

```
Serial.println(rtc.getTimeStr()); //prosedur pembacaan waktu
t = rtc.getTime();
Serial.print("Jam = "); Serial.print(t.hour);
Serial.print(" || Min = "); Serial.print(t.min);
Serial.print(" || Sec = "); Serial.println(t.sec);

// delay (1000); //waktu tunda 1 detik per cycle
}
void Jarak() {
  digitalWrite(Trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);
  timer = pulseIn(Echo, HIGH);
  jarak = timer / 58;
  delay(200);
  Serial.print("Jarak = ");
  Serial.print(jarak);
  Serial.print(" cm");
  Serial.println();
}
```

ESP32CAM

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

const char* ssid = "Inisiapo";
const char* password = "Handava23";

// Initialize Telegram BOT
String BOTtoken = "6259382544:AAFm-x6Q5kFuOcVzAH9HDm95gVdzgM3Boy8"; // your
Bot Token (Get from Botfather)

// Use @myidbot to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
// message you
String CHAT_ID = "6136346142";

bool sendPhoto = false;

WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);

#define FLASH_LED_PIN 4
bool flashState = LOW;

//Checks for new messages every 1 second.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

//CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27

#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define PCLK_GPIO_NUM 22
#define Foto 13
int foto = 0;

void configInitCamera() {
```

```

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; //0-63 lower number means higher quality
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12; //0-63 lower number means higher quality
    config.fb_count = 1;
}

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    delay(1000);
    ESP.restart();
}

// Drop down frame size for higher initial frame rate
sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_CIF); //
UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
}

void handleNewMessages(int numNewMessages) {
    Serial.print("Handle New Messages: ");
    Serial.println(numNewMessages);

    for (int i = 0; i < numNewMessages; i++) {
        String chat_id = String(bot.messages[i].chat_id);
        if (chat_id != CHAT_ID) {
            bot.sendMessage(chat_id, "Unauthorized user", "");
        }
    }
}

```

```

        continue;
    }

    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);

    String from_name = bot.messages[i].from_name;
    if (text == "/start") {
        String welcome = "Welcome , " + from_name + "\n";
        welcome += "Use the following commands to interact with the ESP32-CAM \n";
        welcome += "/photo : takes a new photo\n";
        welcome += "/flash : toggles flash LED \n";
        bot.sendMessage(CHAT_ID, welcome, "");
    }
    if (text == "/flash") {
        flashState = !flashState;
        digitalWrite(FLASH_LED_PIN, flashState);
        Serial.println("Change flash LED state");
    }
    if (text == "/photo") {
        sendPhoto = true;
        Serial.println("New photo request");
    }
    }
}

String sendPhotoTelegram() {
    const char* myDomain = "api.telegram.org";
    String getAll = "";
    String getBody = "";

    camera_fb_t * fb = NULL;
    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println("Camera capture failed");
        delay(1000);
        ESP.restart();
        return "Camera capture failed";
    }

    Serial.println("Connect to " + String(myDomain));

    if (clientTCP.connect(myDomain, 443)) {
        Serial.println("Connection successful");

        String head = "--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"chat_id\";\r\n\r\n" + CHAT_ID + "\r\n--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
        String tail = "\r\n--RandomNerdTutorials--\r\n";

        uint16_t imageLen = fb->len;
        uint16_t extraLen = head.length() + tail.length();
        uint16_t totalLen = imageLen + extraLen;

```

```

clientTCP.println("POST /bot" + BOTtoken + "/sendPhoto HTTP/1.1");
clientTCP.println("Host: " + String(myDomain));
clientTCP.println("Content-Length: " + String(totalLen));
clientTCP.println("Content-Type: multipart/form-data; boundary=RandomNerdTutorials");
clientTCP.println();
clientTCP.print(head);

uint8_t *fbBuf = fb->buf;
size_t fbLen = fb->len;
for (size_t n = 0; n < fbLen; n = n + 1024) {
  if (n + 1024 < fbLen) {
    clientTCP.write(fbBuf, 1024);
    fbBuf += 1024;
  }
  else if (fbLen % 1024 > 0) {
    size_t remainder = fbLen % 1024;
    clientTCP.write(fbBuf, remainder);
  }
}

clientTCP.print(tail);

esp_camera_fb_return(fb);

int waitTime = 10000; // timeout 10 seconds
long startTimer = millis();
boolean state = false;

while ((startTimer + waitTime) > millis()) {
  Serial.print(".");
  delay(100);
  while (clientTCP.available()) {
    char c = clientTCP.read();
    if (state == true) getBody += String(c);
    if (c == '\n') {
      if (getAll.length() == 0) state = true;
      getAll = "";
    }
    else if (c != '\r')
      getAll += String(c);
    startTimer = millis();
  }
  if (getBody.length() > 0) break;
}
clientTCP.stop();
Serial.println(getBody);
}
else {
  getBody = "Connected to api.telegram.org failed.";
  Serial.println("Connected to api.telegram.org failed.");
}
return getBody;
}
void setup() {

```

```

WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
// Init Serial Monitor
Serial.begin(115200);
pinMode(Foto, INPUT_PULLUP);
// Set LED Flash as output
pinMode(FLASH_LED_PIN, OUTPUT);
digitalWrite(FLASH_LED_PIN, flashState);
// Config and init the camera
configInitCamera();
// Connect to Wi-Fi
WiFi.mode(WIFI_STA);
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for
api.telegram.org
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}
Serial.println();
Serial.print("ESP32-CAM IP Address: ");
Serial.println(WiFi.localIP());
}
void loop() {
  foto = digitalRead(Foto);
  Serial.println(foto);
  if (sendPhoto) {
    Serial.println("Preparing photo");
    sendPhotoTelegram();
    sendPhoto = false;
  }
  if (millis() > lastTimeBotRan + botRequestDelay) {
    if (foto == 0) {
      // sendPhoto = true;
      bot.sendMessage(CHAT_ID, "Makanan Akan Habis", "");
      Serial.println("New photo request");
    }
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    while (numNewMessages) {
      Serial.println("got response");
      handleNewMessages(numNewMessages);
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
  }
}
}

```