

Daftar Pustaka

- [1] S. Budiharjo and S. Milah, "KEAMANAN PINTU RUANGAN DENGAN RFID DAN PASSWORD MENGGUNAKAN ARDUINO UNO."
- [2] J. Putra, M. A. Hd, and W. Pamungkas, "SISTEM PENGAMAN PINTU RUMAH MENGGUNAKAN SENSOR RFID RC522 DAN FINGERPRINT BERBASIS INTERNET OF THINGS," vol. 8, no. 2, pp. 14–21, 2022.
- [3] M. Fauza and A. Muthalib, "SISTEM PENGAMAN PINTU OTOMATIS MENGGUNAKAN SENSOR RADIO FREQUENCY IDENTIFICATION (RFID) BERBASIS ARDUINO UNO".
- [4] I. Zaki, "RANCANG BANGUN DOORLOCK SYSTEM DENGAN RFID MENGGUNAKAN ARDUINO."
- [5] E. L. Jurnal and P.-I. Sains, "Rancang Bangun Smart Locker Menggunakan Rfid Berbasis Arduino Uno," vol. 2, no. 1, pp. 2527–6336, 2020.
- [6] P. Arri Ape Pane Basabilik Prodi Fisika, J. Fisika, and U. Tanjungpura, "RANCANG BANGUN SISTEM PEMANTAU KEDATANGAN TAMU BERBASIS INTERNET OF THINGS (IOT)," vol. 9, no. 2, pp. 110–116, 2021.
- [7] "Maullana Ichsanuddin1, Ferdiansyah2".
- [8] J. Evander, "Implementasi Internet of Things Untuk Pembuatan Keyless Smart Door Dengan NFC Tag Dan QR Code."
- [9] S. Sintaro *et al.*, "SISTEM CERDAS SEBAGAI KEAMANAN KANDANG TERNAK SAPI MENGGUNAKAN CAMERA ESP-CAM DAN SELENOID."
- [10] F. Fakhrol Iman, "Purwarupa Smart Door Lock Menggunakan Multi Sensor Berbasis Sistem Arduino."
- [11] A. Purnomo, D. Hardiyanto, and S. Kartikawati, *Smart Doorlock System Menggunakan Kontrol Android Blynk Untuk Pemantauan Keamanan Rumah Tinggal.*
- [12] "FILE III".
- [13] D. Aryani *et al.*, "PERANCANGAN SMART DOOR LOCK MENGGUNAKAN VOICE RECOGNITION BERBASIS RPBERRY PI 3," vol. 4, no. 2, 2018.
- [14] M. Yusup, P. Abas Sunarya, and K. Aprilyanto, "Rancang Bangun Sistem Monitoring Pengukuran Volume Air Berbasis IoT Menggunakan Arduino Wemos," *CERITA*, vol. 6.

- [15] “BAB I PENDAHULUAN.”
- [16] M. I. Pane, H. Dafitri, and I. Faisal, “ANIMASI PERANCANGAN LAMPU LIGHT EMITTING DIODE (LED) PADA TAMAN HIAS A B STRAKARTICLEINFO.”
- [17] L. Hermanto and R. Candra, “SISTEM KEAMANAN BUKA TUTUP KUNCI BRANKAS MENGGUNAKAN SIDIK JARI BERBASIS ARDUINO MEGA SECURITY SYSTEM OF OPENING AND LOCKING SAFE-DEPOSIT BOX USING FINGERPRINTS WITH ARDUINO MEGA-BASE SYSTEM,” 2017.
- [18] H. Jurnal, Y. Nur, I. Fathulrohman, A. Saepuloh, and M. Kom, “JURNAL MANAJEMEN DAN TEKNIK INFORMATIKA ALAT MONITORING SUHU DAN KELEMBABAN MENGGUNAKAN ARDUINO UNO,” *JUMANTAKA*, vol. 02, p. 1, 2018.
- [19] “BAB II LANDASAN TEORI.” [Online]. Available: <https://www.arduino.cc/en/Main/Products>
- [20] Z. Muslimin, M. A. Wicaksono, M. F. Fadlurachman, and I. Ramli, “Rancang Bangun Sistem Keamanan dan Pemantau Tamu pada Pintu Rumah Pintar Berbasis Raspberry Pi dan Chat Bot Telegram,” *Jurnal Penelitian Enjiniring*, vol. 23, no. 2, pp. 121–128, Nov. 2019, doi: 10.25042/jpe.112019.05.
- [21] I. Novansyah, T. B. Utomo, and M. Y. Fadhlan, “Prosiding The 12 th Industrial Research Workshop and National Seminar Bandung,” 2021.

Sensor Pintu Masuk

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

const char* ssid = "Ga ada jaringan";
const char* password = "bentarbentar";
#define BOTtoken
    "5878608621:AAGWFSn8tx4cdMXlyqJRL2GW6rnXmvjCg24" // your
    Bot Token (Get from Botfather)
#define CHAT_ID "1214046787"

//const char* ssid = "Ga ada jaringan";
//const char* password = "bentarbentar";
//#define BOTtoken
    "5878608621:AAGWFSn8tx4cdMXlyqJRL2GW6rnXmvjCg24" // your
    Bot Token (Get from Botfather)
//#define CHAT_ID "1214046787"

WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

byte sda = 21;
byte rst = 22;
#include<KRRfid.h>

const int buzzer = 26;
const int relay = 25;
const int led1 = 12;
```

The image contains a large watermark of the Universitas Nasional logo, which is a shield-shaped emblem with a yellow border. Inside the shield, there is a green field with a white and red stylized 'U' shape. At the top of the shield is a yellow five-pointed star. The text 'UNIVERSITAS NASIONAL' is written in white capital letters along the bottom edge of the shield.

```

const int led2 = 14;

int stt = 0, count = 0, sensorDor;
int stl = 0;
String nama = "";
boolean stb = false, stcap = false;

#include <HardwareSerial.h>
HardwareSerial SerialPort(2); // use UART2

const int dorSensor = 15;
const int Touch = 2;
int touch;

void setup() {
  Serial.begin(115200);
  rfidBegin();
  SerialPort.begin(115200, SERIAL_8N1, 16, 17); //17 ke 14... 16 ke 15
  WiFi.mode(WIFI_STA);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print('.');
  }
  client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root
  certificate for api.telegram.org
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

```

```
pinMode(buzzer, OUTPUT);
pinMode(dorSensor, INPUT_PULLUP);
pinMode(Touch, INPUT);
pinMode(relay, OUTPUT);
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
digitalWrite(relay, HIGH);
}

void loop() {
  sensordor = digitalRead(dorSensor);
  touch = digitalRead(Touch);
  getTAG();
  if (TAG != "") {
    Serial.print("TAG :");
    Serial.println(TAG);
    if (TAG == "73211238") {
      digitalWrite(led2, HIGH);
      digitalWrite(relay, LOW);
      nama = "Agus";
      stt = 1;
      stl = 1;
      count = 0;
      bmasuk();
    } else if (TAG == "111111") {
      digitalWrite(led2, HIGH);
      digitalWrite(relay, LOW);
      nama = "Bamabang";
      stt = 1;
      stl = 1;

```



```
count = 0;
bmasuk();
}
else {
digitalWrite(led1, HIGH);
digitalWrite(buzzer, HIGH);
delay(400);
digitalWrite(buzzer, LOW);
delay(100);
nama = "Tidak Di Ketahui Yang Tap Kartu";
stt = 1;
}
TAG = "";
}

if (touch == 1) {
digitalWrite(led2, HIGH);
digitalWrite(relay, LOW);
nama = "Keluar";
stt = 1;
stl = 1;
count = 0;
bmasuk();
}

if (stt == 1) {
SerialPort.print(1);
bot.sendMessage(CHAT_ID, "Masuk An :" + String(nama), "");
stt = 0;
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
```

```
}

if (stl == 1) {
  count++;
  if (count > 10) {
    count = 11;
    digitalWrite(relay, HIGH);
    stl = 0;
  }
  Serial.println("count= " + String(count));
}
else {
  count = 0;
}
Serial.println("sensordor= " + String(sensordor));
Serial.println("touch= " + String(touch));
Serial.println("stl= " + String(stl));
Serial.println("stt= " + String(stt));

if (sensordor == 1 && stl == 0) {
  digitalWrite(led1, HIGH);
  digitalWrite(buzzer, HIGH);
  if (stcap == false) {
    SerialPort.print(1);
    bot.sendMessage(CHAT_ID, "Pintu di Buka Secaraa paksa!!!", "");
    stcap = true;
  }
}
else {
  stcap = false;
}
```



```
digitalWrite(led1, LOW);  
digitalWrite(buzzer, LOW);  
}  
delay(1000);  
}
```

```
void bmasuk() {  
digitalWrite(buzzer, HIGH);  
delay(500);  
digitalWrite(buzzer, LOW);  
delay(500);  
digitalWrite(buzzer, HIGH);  
delay(500);  
digitalWrite(buzzer, LOW);  
delay(100);  
}
```



Capture Telegram

```
#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 = "Ga ada jaringan";
const char* password = "bentarbentar";

// Initialize Telegram BOT
String BOTtoken =
"5878608621:AAGWFSn8tx4cdMXlyqJRL2GW6rnXmvjCg24"; // your Bot
Token (Get from Botfather)
String CHAT_ID = "1214046787";

bool sendPhoto = false;

WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);

#define FLASH_LED_PIN 4
bool flashState = LOW;

const int led = 2;

//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
```

```
#include <HardwareSerial.h>
```

```
HardwareSerial SerialPort(1); // use UART2
```

```
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;
}

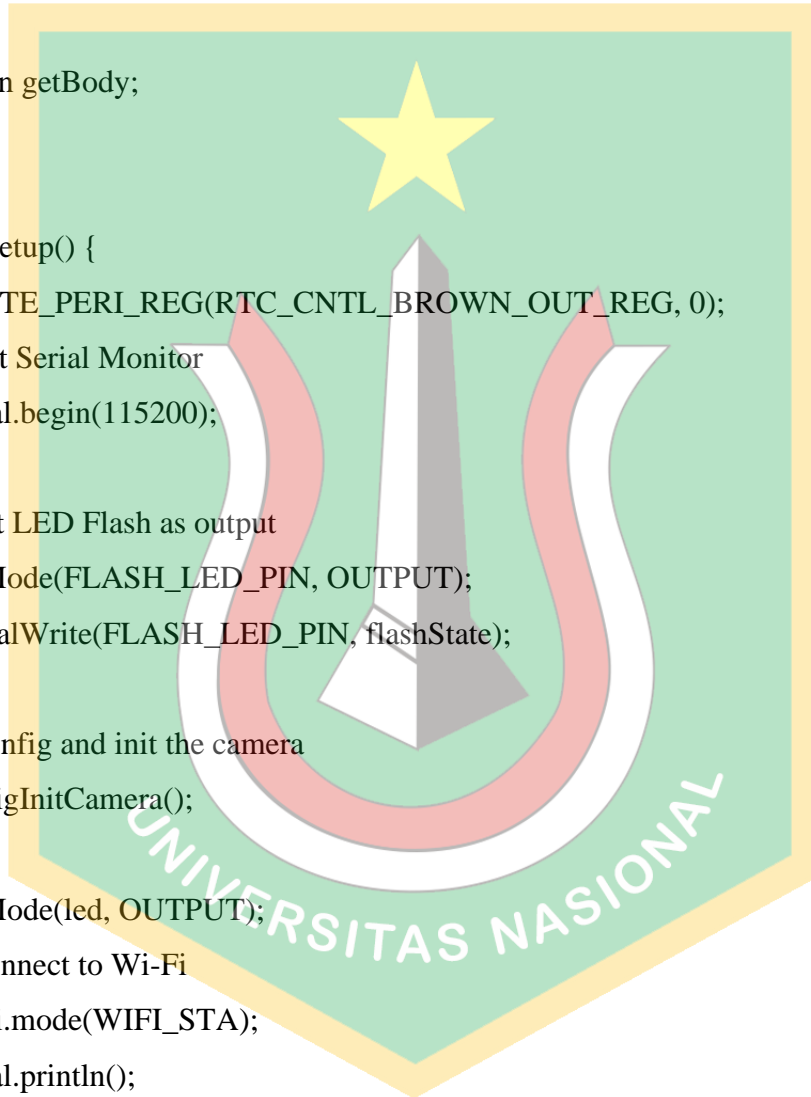
void setup() {
    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
    // Init Serial Monitor
    Serial.begin(115200);

    // Set LED Flash as output
    pinMode(FLASH_LED_PIN, OUTPUT);
    digitalWrite(FLASH_LED_PIN, flashState);

    // Config and init the camera
    configInitCamera();

    pinMode(led, OUTPUT);
    // 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) {

```



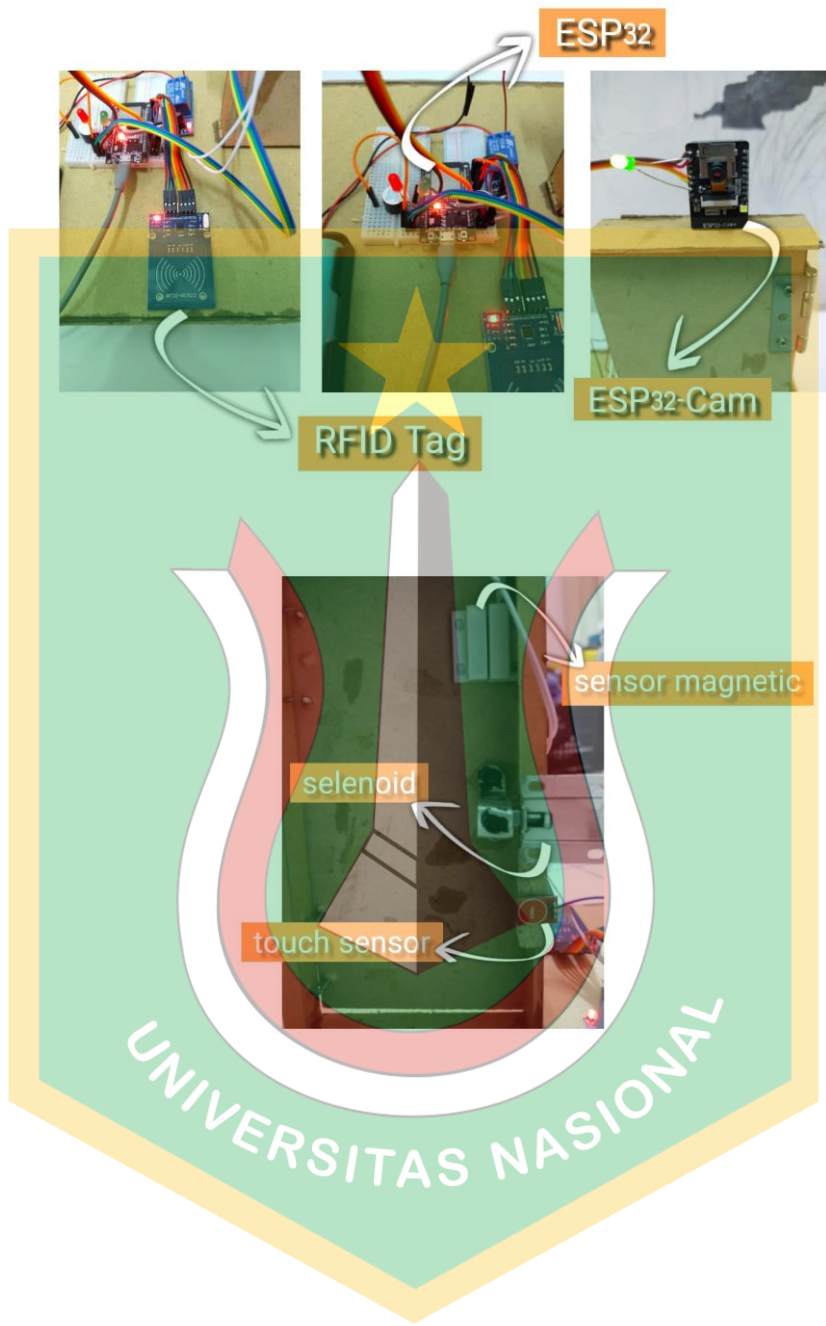

```
Serial.print(".");
delay(500);
}
Serial.println();
Serial.print("ESP32-CAM IP Address: ");
Serial.println(WiFi.localIP());
SerialPort.begin(115200, SERIAL_8N1, 14, 15);
}

void loop() {

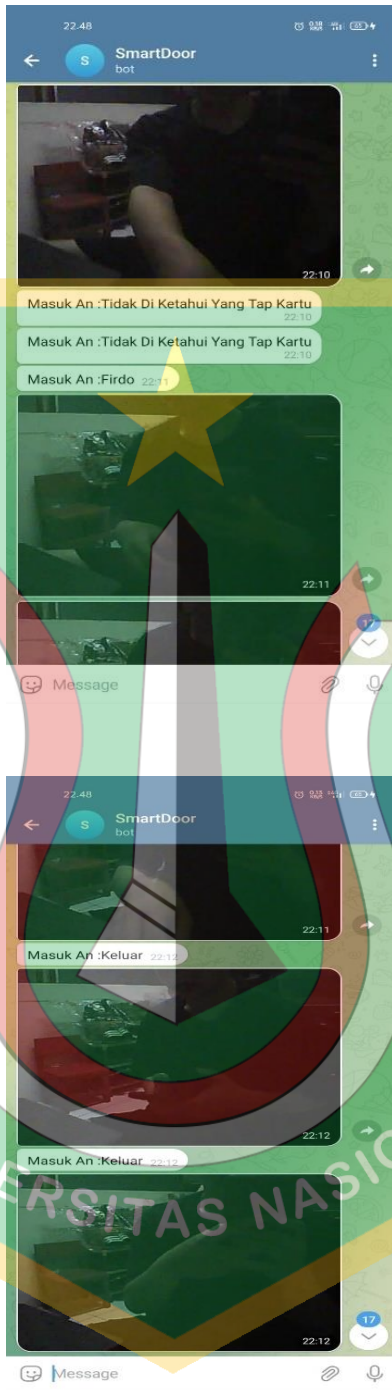
digitalWrite(led, HIGH);
if (Serial.available())
{

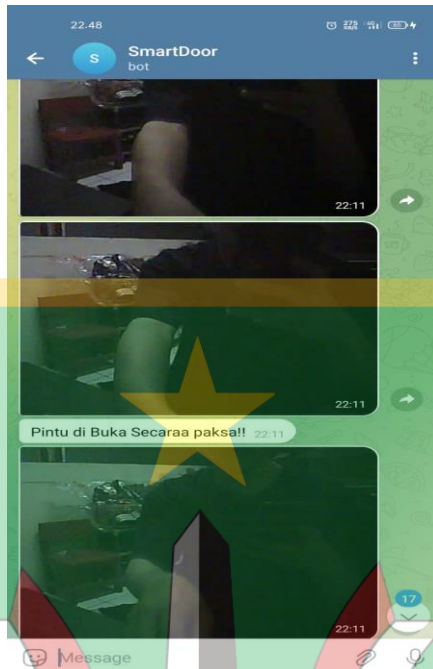
char number = Serial.read();
if (number == '1') {
sendPhotoTelegram();
}
Serial.println(number);
number = ' ';
}
}
```











Skripsi_FirdoAndriSaputra_Informatika

ORIGINALITY REPORT

17 %	16%	3%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.iain-surakarta.ac.id Internet Source	1 %
2	ejournal.borobudur.ac.id Internet Source	1 %
3	docplayer.info Internet Source	1 %
4	cot.unhas.ac.id Internet Source	1 %
5	eprints.polsri.ac.id Internet Source	1 %
6	eprints.uty.ac.id Internet Source	1 %
7	digilib.uinsby.ac.id Internet Source	1 %
8	repo.darmajaya.ac.id Internet Source	1 %
9	repository.unib.ac.id Internet Source	1 %
10	eprints.uniska-bjm.ac.id Internet Source	1 %
11	Submitted to Institut Teknologi Nasional Malang Student Paper	1 %
12	Submitted to Universitas Budi Luhur Student Paper	1 %
13	repository.uir.ac.id Internet Source	1 %
14	www.jogjarobotika.com Internet Source	<1 %
15	ejurnal.teknokrat.ac.id Internet Source	<1 %
16	www.coursehero.com Internet Source	<1 %
17	adoc.pub Internet Source	<1 %
18	repository.unsri.ac.id Internet Source	<1 %
19	digilib.uns.ac.id Internet Source	<1 %
20	repository.its.ac.id Internet Source	<1 %
21	repository.usd.ac.id	