



## DAFTARPUSTAKA

## DAFTARPUSTAKA

Aditya, Mochamad Ari. (2020). SISTEM INFORMASI KEAMANAN KANDANG KAMBING BERBASIS INTERNET OF THINGS. Universitas Muhammadiyah Gresik

Alexander Sinaga, Aswardi. (2020). Rancangan Alat Penyiram Dan Pemupukan Tanaman Otomatis Menggunakan RTC Dan Soil Moisture Sensor Berbasis Arduino. Universitas Negeri Padang.

Budhi Sumboro, Sutariyani, Ronny Imam Utomo. (2020). Sistem Keamanan Rumah Berbasis Raspberry Pi dan Menggunakan Sensor PIR. STMIK AUB, Surakarta, Indonesia

Budiharto, Widodo. (2010). Robotika Teori dan Implementasi. Yogyakarta: Penerbit Andi Offset.

### TECHNOLOGY AND ELECTRONICS

<http://electronicsbot.blogspot.com/2019/06/module-ftdi-serialtottl.html> . Diakses pada tanggal 22 Juni 2021. Pukul 23:45 WIB

Muhammad Rizky Kaisupy. (2017). PENGEMBANGAN IMPLEMENTASI SISTEM MONITORING DAN KONTROL OTOMATIS SUHU DAN KELEMBABAN PADA BUDIDAYA JAMUR MENGGUNAKAN NI MYRIO-1900. UNIVERSITAS MUHAMMADIYAH MALANG.

Nanda Tommy Wirawan. (2018). PEMANFAATAN SMARTPHONE PADA ROBOT BERODA UNTUK MONITORING JARAK ROBOT DENGAN HALANGAN MENGGUNAKAN BLUETOOTH HC-05 SEBAGAI MEDIA KOMUNIKASI. LPPM UPI YPTK Padang

Nurjamila, Gebri Yuni sini, Adrian Agus Prasetya. (2020). RANCANG BANGUN APLIKASI HADITS BUKHARI BERBASIS ANDROID. STMIK Kreatindo Manokwari.

Rena Sahani Dian S. dan Fidelis Agus Priyambodo. (2014). RANCANG BANGUN SISTEM PENGUNCI OTOMATIS DENGAN KENDALI AKSES MENGGUNAKAN RFID CARD DAN PASSWORD BERBASIS MIKROKONTROLER ATMEGA 16.

Rini Suwartika, Gandang Sembada. (2020). Perancangan Sistem Keamanan Menggunakan Solenoid Door Lock Berbasis Arduino Uno pada Pintu Laboratorium di PT. XYZ. Politeknik Piksi Ganesha Bandung, Bandung, Indonesia.

Risky Arianto. (2020). RANCANG BANGUN TIMBANGAN DIGITAL DENGAN WIRELESS DISPLAY. UNIVERSITAS PGRI ADI BUANA SURABAYA.

Selia Kijany Br Ginting. (2019). RANCANG BANGUN LOKER OTOMATIS DENGAN MEMANFAATKAN RFID CARD DAN BLUETOOTH BERBASIS MIKRIKONTROLLER ATMEGA328.




Unipa Surabaya








UNIVERSITAS PGRI ADI BUANA SURABAYA FAKULTAS TEKNIK  
Program Studi : Teknik Lingkungan – Perencanaan Wilayah Kota  
Teknik Industri – Teknik Elektro - PVKK  
KAMPUS II: Jl. Dukuh Menanggal XII/4 ☎ (031) 8281181 Surabaya 60234  
Website : [www.ft.unipasby.ac.id](http://www.ft.unipasby.ac.id) E-mail : [ft@unipasby.ac.id](mailto:ft@unipasby.ac.id)

## BERITA ACARA BIMBINGAN SKRIPSI

Form Skripsi-03

Nama	: Muhammad Arelri Eko Prasetyo	
NIM	: 173600050	
Program Studi	: Teknik Elektro	
Pembimbing	DRS. WIDODO, S.T, M.KOM. IR. WINARNO F.B, M.ENG.	
Periode Bimbingan	: <del>Genap</del> /Genap*) Tahun 2020 / 2021	
Judul Skripsi	Rancang Bangun Sistem Keamanan Loker Berbasis Internet of Things (IoT) Interface Android	

### KEGIATAN KONSULTASI / BIMBINGAN

No	Tanggal	Materi pembimbingan	Keterangan	Paraf
1	24-03-2021	Bab I Pendahuluan	Acc	
2	22-04-2021	Bab II Kajian Pustaka	Acc	
3	29-05-2021	Bab III Metode penelitian	Acc	
4	05-06-2021	Rancang Alat, kendala dan solusi	Acc	
5	14-06-2021	Menganalisis data	Acc	
6	16-06-2021	Bab IV hasil dan pengujian data	Acc	
7	18-06-2021	Evaluasi alat	Revisi	

B	23-06-2021	mengambil kesimpulan dan saran	Acc	8
Dinyatakan selesai tanggal : ...24 Juni..... 20.21				

Mengetahui,  
Ketua Program Studi,



Akbar Sujiwa, S.SI., M.SI

Pembimbing,

A handwritten signature in black ink, appearing to read "Widodo".

Drs. Widodo, S.T, M.Kom

Surabaya, 24 Juni 2021  
Mahasiswa,

A handwritten signature in black ink, appearing to read "Muhammad Andri Eko Prasetyo".

Muhammad Andri Eko Prasetyo



# UNIVERSITAS PGRI ADI BUANA SURABAYA

## FAKULTAS TEKNIK

Program Studi : Teknik Lingkungan – Perencanaan Wilayah Kota  
Teknik Industri – Teknik Elektro - PVKK

KAMPUS II: Jl. Dukuh Menanggal XII/4 ☎ (031) 8281181 Surabaya 60234

Website : [www.ft.unipasby.ac.id](http://www.ft.unipasby.ac.id) E-mail : [ft@unipasby.ac.id](mailto:ft@unipasby.ac.id)

### FORM REVISI SKRIPSI

Nama Mahasiswa : Muhammad Andri Eko Prasetyo  
NIM : 1736 00050  
Fakultas / Progdi : Teknik / Teknik Elektro  
Judul Skripsi : Rancang Bangun Sistem Keamanan Loker Berbasis Internet of Things (IoT) Interface Android

Ujian Tanggal : 29 Juni 2021

No Bab.	Tanggal	Materi Konsultasi	Keterangan Catatan	Tanda Tangan Penguji
I	9 Juli 2021	Penulisan kata pengantar		
II				
III	10 Juli 2021	flowchart dikasih keterangan		
IV	20 Juli 2021	Grafik di pisah		
V				

Disetujui Dosen Penguji

Pada Tanggal,.....

Penguji I,

(Atmiasri, S.T., M.T.)

Penguji II,

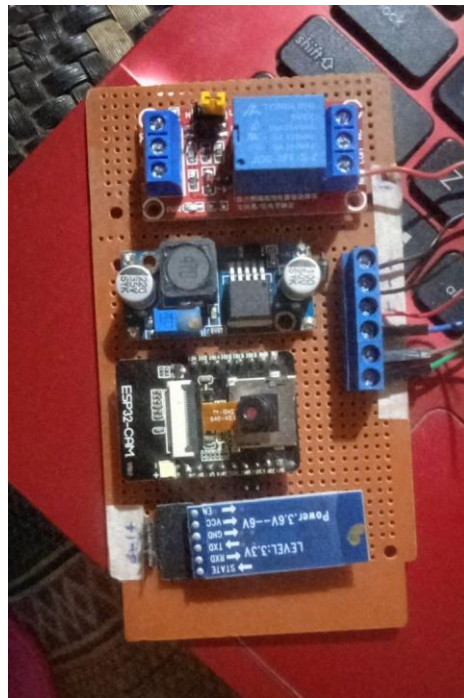
(Akbar Sujiwa, S.Si., M.Si.)

- Penyelesaian Revisi paling lambat 2 minggu dari pelaksanaan Ujian Skripsi.
  - Pengetikan, penjilidan, penandatanganan Skripsi dan mengumpulkan Skripsi paling lambat 2 minggu dari revisi.
- Apabila sampai batas waktu tersebut ( point 1, a dan b ) mahasiswa belum menyelesaikan revisidan tanda tangan, maka **Ujian dinyatakan Gugur.**
- Foto copy Form Revisi diserahkan ke Program Studi.
  - Skripsi yang sudah direvisi diserahkan ke Fakultas tiga eksemplar untuk dijilid.

Lampiran 1. Gambar Alat



Gambar Loker



Gambar Rangkaian

Lampiran 2. Gambar Hasil Pengujian Telegram



Gambar Hasil Pengujian Telegram

### Lampiran 3. Hasil Pengujian Sensor PIR



Gambar Pengujian Sensor PIR

### Lampiran 4. Program



```
#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>

#include <Wire.h>

const char* ssid = "manusia"; //ssid wifi const char*

password = "1sampai7";//password wifi

String chatId = "1455299198"; //chat id telegram

String BOTtoken = "1691435650:AAEFgouzk43tQ9VXEZMtZHCjkDhFSe5_BB4"; //bot
token telegram

bool sendPhoto = false;

WiFiClientSecure clientTCP;

UniversalTelegramBot bot(BOTtoken, clientTCP);

//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 FLASH_LED_PIN    4          //pin flash esp32-cam
#define pir      13      //pin sensor pir #define sel_lock 2

//pin relay doorlock

bool flashState =
LOW; bool flag = 0;

bool detect = 0;

int botRequestDelay = 1000; // mean time between scan messages long

lastTimeBotRan; // last time messages' scan has been done

```

```

void handleNewMessages(int numNewMessages);

String sendPhotoTelegram();

char receivedChar;// received value will be stored as CHAR in this variable

const char openDL ='b'; const char closeDL ='a'; bool security = false;

#####

void setup(){

    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);

//    Serial.begin(115200);

    Serial.begin(9600);

    pinMode(FLASH_LED_PIN, OUTPUT);

    pinMode(pir,INPUT_PULLUP);

    digitalWrite(FLASH_LED_PIN, flashState);

    WiFi.mode(WIFI_STA);

    Serial.println();

    Serial.print("Connecting to ");

    Serial.println(ssid);

    WiFi.begin(ssid, password); while

(WiFi.status() != WL_CONNECTED) {

```

```

Serial.print(".");

delay(500);

}

Serial.println();

Serial.print("ESP32-CAM IP Address: ");

Serial.println(WiFi.localIP());

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_SVGA);          //
UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA

##### BLUETOOTH #####

Serial.println("To Close Doorlock send: a");//print on serial monitor

Serial.println("To Open Doorlock send: b");//print on serial monitor

pinMode(sel_lock, OUTPUT);

#####

}

void loop() { if

    (Serial.available()>0) {

        ControlBluetooth();    //control doorlock bluetooth

    }

else { while (WiFi.status() !=

    WL_CONNECTED) {

```

```

WiFi.begin(ssid, password);

Serial.print(".");

delay(2000);

}

if (sendPhoto){           //mengambil foto

//      Serial.println("Preparing      photo");

  bot.sendMessage(chatId, "Mengambil foto", "");

  sendPhotoTelegram(); sendPhoto = false;

}

if (flag){                //sensor pir aktif

//      delay(1000);

  if(detect == 0){

    digitalRead(pir);

    MotionDetected(); //mengirim foto hasil deteksi gerakan

  }

}

if (millis() > lastTimeBotRan + botRequestDelay){ 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();

    }

// delay(850); delay(500);

    }

}

void MotionDetected() { if

(digitalRead(pir)== 1) {

detect = 1;

    Serial.print("Motion Detected, Value = ");

    Serial.println(digitalRead(pir));

    String motion = "Terdeteksi gerakan!!\n";

    motion += "Foto akan segera dikirim\n";

    bot.sendMessage(chatId, motion, "");

    sendPhotoTelegram(); delay(1000);

    //tambahan detect = 0;

    }

else {}

}

```



```

void(*reset_esp32) (void) = 0; //fungsi reset

void ControlBluetooth() {

// if (Serial.available(>0) {

    receivedChar =(char)Serial.read();

    Serial.print ("Received:"); //print
    on serial monitor

    Serial.println(receivedChar);

        //print on serial monitor

    if(receivedChar == closeDL)

    {

        flag = 1;

        Serial.println("Locked "); //write on serial monitor

        digitalWrite(sel_lock, LOW); //mengunci doorlock

    }

    else if(receivedChar == openDL)

    {

        flag = 0;

```

```

        Serial.println("UnLocked");    //write on serial monitor

        digitalWrite(sel_lock, HIGH); //membuka doorlock

    }

    else if(receivedChar == 'c')

    {

        reset_esp32();                //mereset esp32-cam

    }

//    }

    delay(20);

}

```

```

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" + chatId + "\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 = 1000;    // 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 (c == '\n'){ if

    (getAll.length()==0) state=true;

        getAll = "";

    }

    else if (c != '\r'){

        getAll +=

        String(c);

    }

    if (state==true){

        getBody +=

        String(c);

    }

    startTimer = millis();

    }

    if (getBody.length(>0) break;

}

clientTCP.stop();

//    Serial.println(getBody);

```

```

        Serial.println();
    }

    else { getBody="Connected to api.telegram.org

        failed.";

        Serial.println("Connected to api.telegram.org failed.");

    }

    return getBody;

}

void handleNewMessages(int numNewMessages){

    Serial.print("Handle New Messages: ");

    Serial.println(numNewMessages);

    for (int i = 0; i < numNewMessages; i++){

        // Chat id of the requester

        String chat_id = String(bot.messages[i].chat_id); if

        (chat_id != chatId){ bot.sendMessage(chat_id,

        "Unauthorized user", ""); continue;

    }

    // Print the received message

    String text = bot.messages[i].text;

    Serial.println(text);

```

```
String fromName = bot.messages[i].from_name;
```

```
if (text == "/flash"){ flashState =
```

```
    !flashState;
```

```
    digitalWrite(FLASH_LED_PIN,
```

```
    flashState);
```

```
}
```

```
if (text == "/photo"){
```

```
    sendPhoto = true;
```

```
    Serial.println("New photo request");
```

```
}
```

```
if (text == "/piron"){
```

```
    flag = 1;
```

```
        bot.sendMessage(chatId, "PIR Sensor sudah aktif, Saat terjadi gerakan anda akan  
dikirimkan foto", "");
```

```
}
```

```
if (text == "/piroff"){
```

```
    flag = 0;
```

```
        bot.sendMessage(chatId, "PIR sensor sudah mati, Anda tidak akan menerima pemberitahuan lagi saat terjadi gerakan", "");
```

```
    }
```

```
if (text == "/start"){
```

```
    String welcome = "Selamat datang di ESP32-CAM Telegram Bot bosku .\n";
```

```
    welcome += "Berikut adalah hal yang dapat anda lakukan :\n";
```

```
    welcome += "/photo : Ambil foto\n"; welcome += "/flash :
```

```
Menyalakan/Mematikan flash LED\n"; welcome += "/piron :
```

```
Mengaktifkan sensor PIR\n"; welcome += "/piroff :
```

```
Mematikan sensor PIR\n";
```

```
        welcome += "Anda juga akan mendapatkan notifikasi saat terjadi gerakan dari sensor PIR\n"; bot.sendMessage(chatId, welcome, "Markdown");
```

```
    }
```

```
}
```

```
}
```