



**MANUAL BOOK ARDUINO HOME SECURITY SYSTEM
PRAKTIKUM RANCANGAN DAN APLIKASI IOT**

Disusun Oleh :

1. Arfian Caya Dwi Setya (17/410650/SV/12577)
2. Dewi Hajar (17/410653/SV/12580)

**PROGRAM SARJANA TERAPAN TEKNOLOGI REKAYASA INTERNET
SEKOLAH VOKASI
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2019**

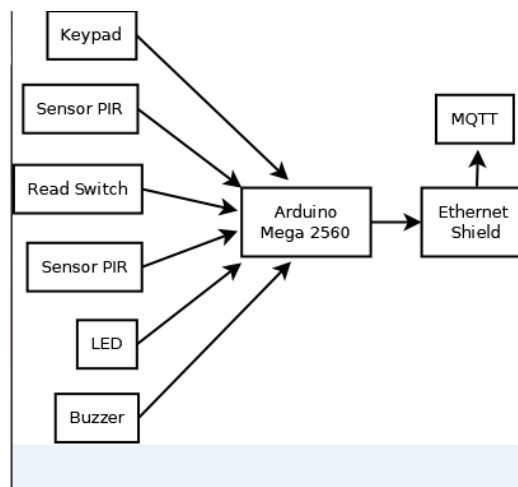
Sistem keamanan rumah dengan menggunakan Arduino Mega 2560, yang akan memicu alarm yang dihasilkan dari sensor pir ketika pintu dibuka atau terdeteksi pergerakan di dalam ruangan ketika sistem sedang aktif.

Tahapan Pertama : Barang yang dibutuhkan

1. Arduino Mega 2560, komponen utama dalam system ini
2. Kabel Jumper, untuk koneksi maupun penyambungan koneksi
3. Breadboard, untuk meletakkan atau menggabungkan komponen dan koneksi
4. Sensor PIR, mendeteksi pergerakan yang terdapat didalam ruang
5. Magnetic Switch, melekat pada pintu untuk mendeteksi pintu ketika dalam kondisi terbuka atau tertutup
6. LCD dan I2C, untuk menampilkan hasil dan menunjukkan lokasi gerakan yang terdeteksi, status alarm, dll
7. Digital Pad, untuk memasukkan pin saat system ingin diaktifkan atau dimatikan
8. Buzzer sebagai pemberi tanda bunyi (alarm) saat terjadi pergerakan ketika motion dinyalakan.

Tahapan Kedua : Diagram Blok Sistem

Dengan diagram blok sistem maka pembuatan alat ini akan tertata dan sesuai dengan konsep awal yang akan kami rancang. Serta dapat mengetahui cara kerja dan penempatan pin serta kesalahan yang nantinya terjadi.

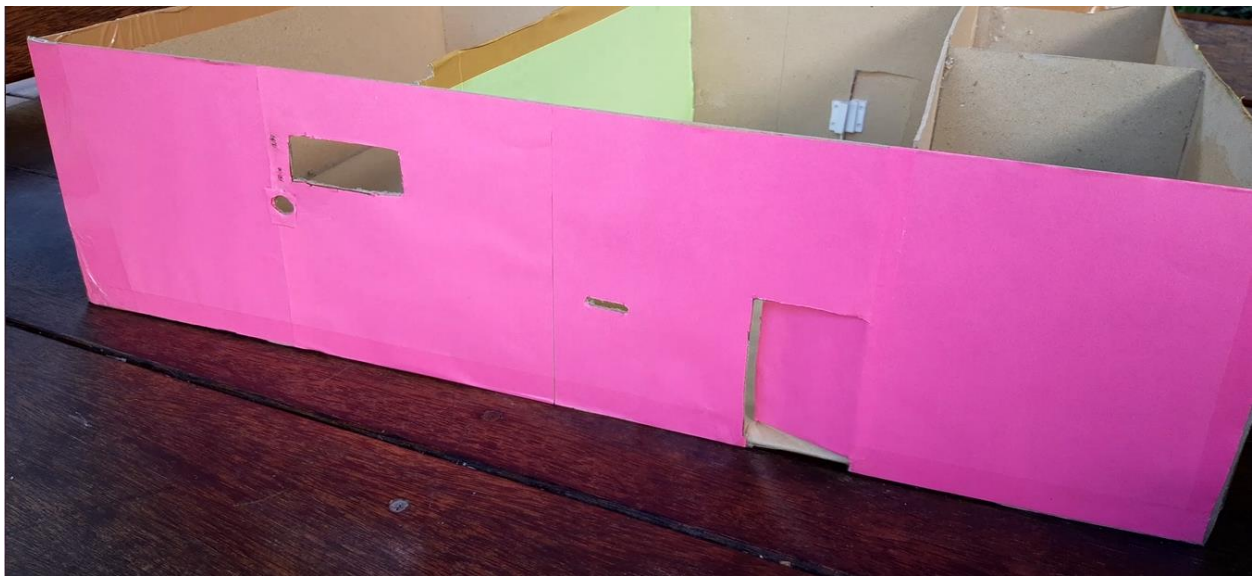


Tahapan Ketiga : Koneksi Perangkat

Pertama yang akan diuji yaitu tampilan layer pada LCD dengan kode pendek, menguji fungsionalitas dari setiap komponen untuk menyelesaikan masalah kesalahan dan menghindari sistem yang tidak dapat berfungsi. Sebagian besar masalah yang akan timbul disebabkan oleh kesalahan kode atau komponen yang rusak.

Tahapan Keempat : Membuat Prototype

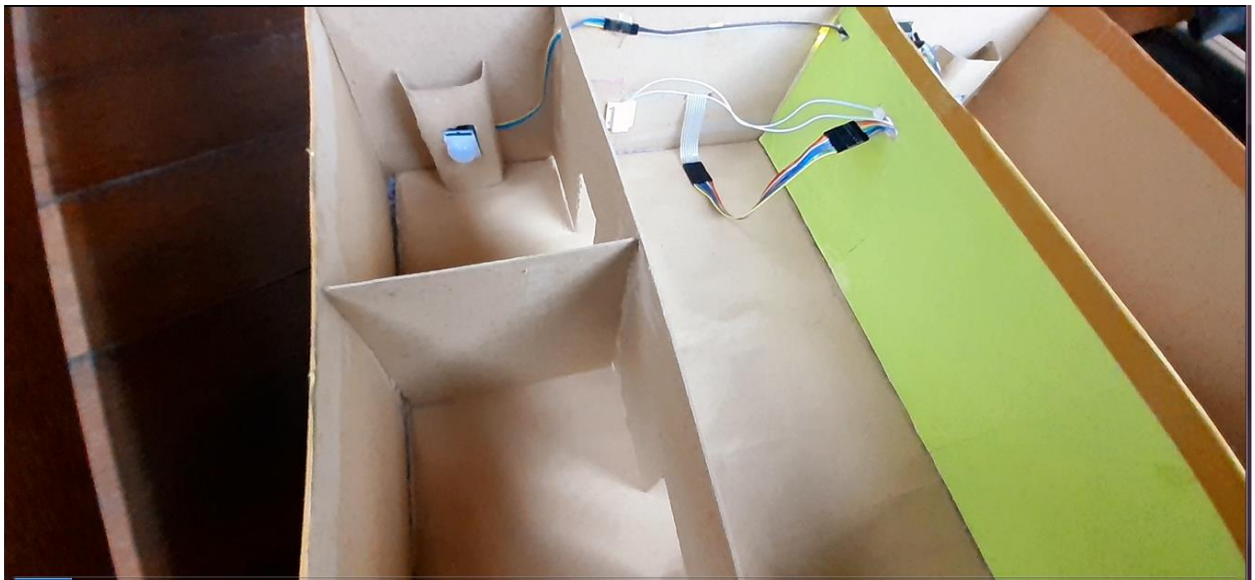
Membuat prototype rumah tergantung dengan model yang diinginkan, langkah pertama yaitu mendesain dan membangun, yang akan membantu untuk menguji perangkat dalam situasi waktunya. Kami menggunakan duplex untuk membuat prototype rumah ini, disini akan dibangun 2 kamar, 1 server serta satu ruang tengah yang berada tepat dibelakang pintu depan rumah. Disini kami membutuhkan 2 sensor PIR dan kode tambahan.

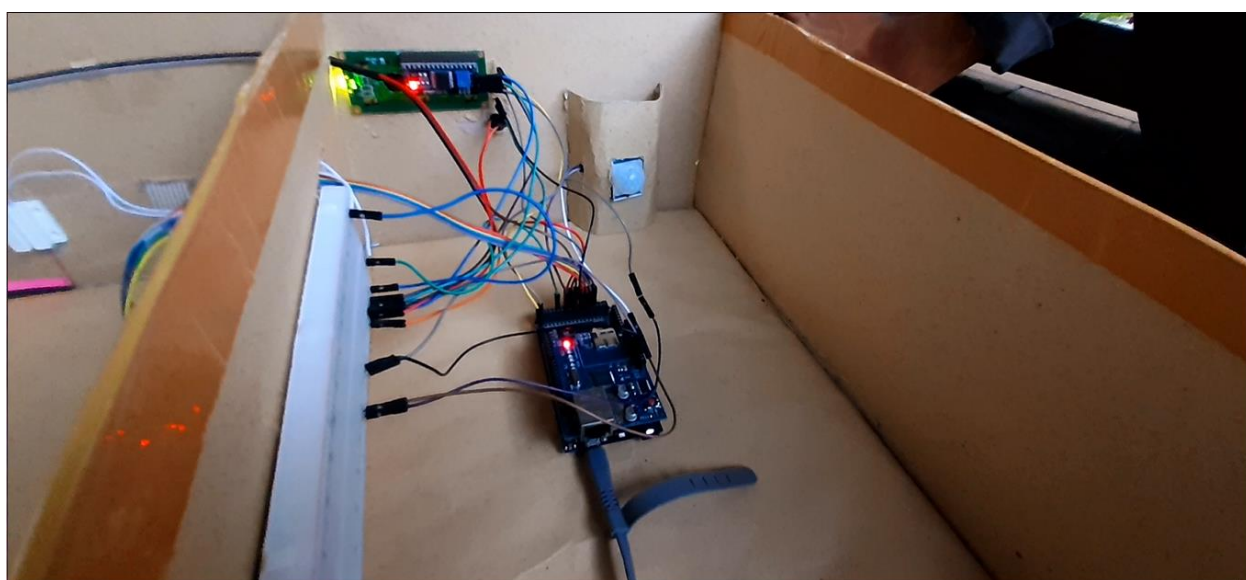
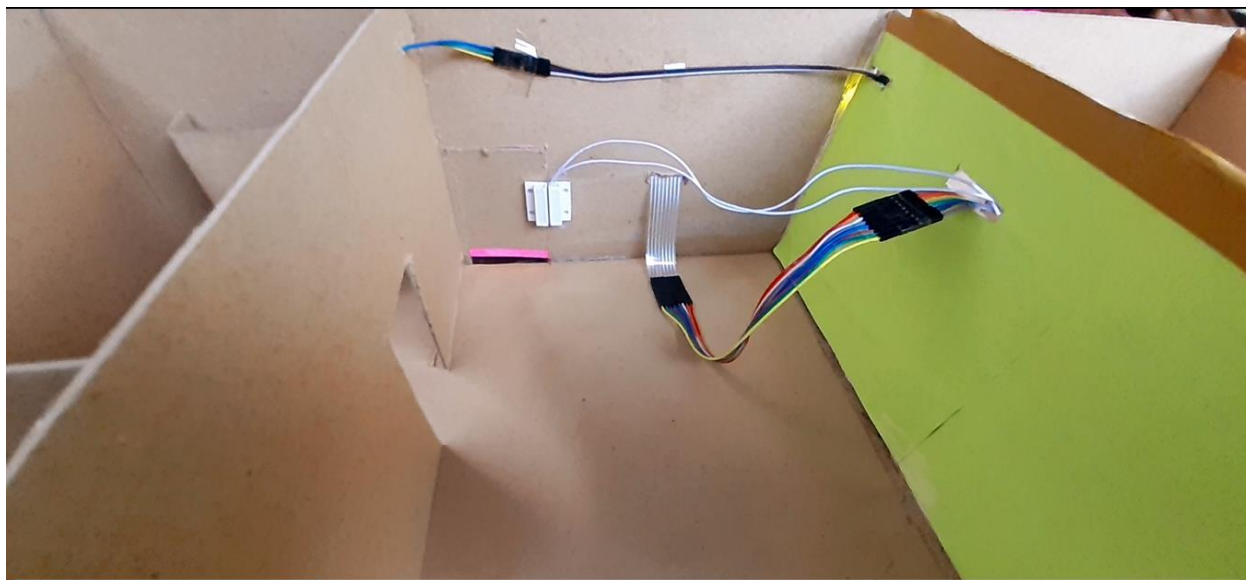




Tahapan Kelima : Perakitan Komponen ke Prototype

Magnetic Switc Door terdapat pada pintu bagian depan prototype, sedangkan sensor PIR berada pada ruang server dan berada di kamar.





Tahapan Keenam : Pengujian

```
////////////////////////////////////  
#include <SPI.h>  
#include <Ethernet.h>  
#include <PubSubClient.h>  
#include <LiquidCrystal_I2C.h>  
#include <Wire.h>  
#include <Password.h> //http://www.arduino.cc/playground/uploads/Code/Password.zip  
#include <Keypad.h> //http://www.arduino.cc/playground/uploads/Code/Keypad.zip  
#include <Servo.h>  
#include "RTCLib.h"  
  
//Ethernet  
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //physical mac address  
IPAddress dnServer(10, 13, 10, 13);  
//IPAddress gateway(10, 33, 107, 254);  
IPAddress gateway(10, 39, 52, 254);  
IPAddress subnet(255, 255, 255, 0);  
IPAddress server(10, 33, 109, 82);  
//IPAddress ip(10, 33, 107, 179);  
IPAddress ip(10, 39, 52, 145);  
String readString;  
  
//Servo  
Servo myservo;    // create servo object to control a servo  
int pos = 90;     // variable to store the servo position  
int passwd_pos = 12; // the position of the password input
```

```
//Password

Password password = Password( "1" );

const byte ROWS = 4; // Four rows

const byte COLS = 4; // Four columns

char keys[ROWS][COLS] = { // Define the Keymap

    {
        '1','2','3','A'    }

    ,

    {
        '4','5','6','B'    }

    ,

    {
        '7','8','9','C'    }

    ,

    {
        '*', '0', '#', 'D'  }

};

byte rowPins[ROWS] = {
    44, 42, 45, 43}; //connect to the row pinouts of the keypad

byte colPins[COLS] = {
    49, 47, 48, 46}; //connect to the column pinouts of the keypad

// Create the Keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

// initialize the library with the numbers of the interface pins

LiquidCrystal_I2C lcd(0x27, 16, 2); // Assignign arduino pins to LCD display module
```

```
//Stroke LED Lights
//int ledDelay = 50; // delay by 50ms
int redPin = 29;
//int bluePin = 30;
//constants for LEDs, inputs and outputs
//int blueLED = 36;
int greenLED = 38;
int redLED = 37;
int pirPin1 = 39;
int pirPin2 = 34;
int reedPin1 = 31;
int reedPin2 = 33;
int speakerPin = 35;
int alarmStatus = 0;
int zone = 0;
int alarmActive = 0;
const char *TOPIC = "bbb";
void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
    Serial.println();
}
```



```
EthernetClient ethClient;
PubSubClient client(ethClient);
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("arduinoClient")) {
      client.subscribe(TOPIC);
      Serial.println("connected");
      Serial.print("Publishing to: ");
      Serial.println(TOPIC);
      // Once connected, publish an announcement...
      //client.publish("kucing", "test");
      // ... and resubscribe
      //client.subscribe("topic/Kamu");
      //Serial.println(client.subscribe("Kamu"));
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

```
void setup(){
  Serial.begin(9600);

  lcd.begin(16, 2);

  //Adding time
  Wire.begin();
  RTC.begin();

  Ethernet.begin(mac, ip, dnServer, gateway, subnet);

  //If we remove the comment from the following line, we will set up the module time and date with
  the computer one
  //RTC.adjust(Date
  //Time(_DATE, __TIME_));
  myservo.attach(2); // attaches the servo on pin 2 to the servo object
  client.setServer(server, 1883);
  client.setCallback(callback);
  displayCodeEntryScreen();
  //Police LED Lights
  pinMode(redPin, OUTPUT);
  // pinMode(bluePin, OUTPUT);
  //setup and turn off both LEDs
  pinMode(redLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(speakerPin, OUTPUT);
  //pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT); //12V Blue LED lighting
  pinMode(pirPin1, INPUT); //Bedroom 2
  pinMode(pirPin2, INPUT); //Garage
  pinMode(reedPin1, INPUT); //Front door
  pinMode(reedPin2, INPUT); //Back door
```

```
digitalWrite(redLED, LOW);
digitalWrite(greenLED, HIGH);
digitalWrite(speakerPin, LOW);
//digitalWrite(relay1, LOW); //
digitalWrite(relay2, HIGH); // 12V Blue LED lighting
keypad.addEventListener(keypadEvent); //add an event listener for this keypad
myservo.write(pos);
delay(500);
}
void loop(){
  keypad.getKey();
  //Serial.println(digitalRead(reedPin1));
  reconnect();
  //Serial.println(digitalRead(pirPin1));
  //Serial.println(digitalRead(pirPin2));
  if (alarmActive == 1){
    if (digitalRead(pirPin1) == HIGH)
    {
      zone = 0;
      alarmTriggered();
    }
    if (digitalRead(reedPin1) == HIGH)
    {
      zone = 1;
      alarmTriggered();
    }
    if (digitalRead(reedPin2) == LOW)
    {
```

```

    zone = 2;

    alarmTriggered();
}

if (digitalRead(pirPin2) == HIGH)
{
    zone = 3;
    alarmTriggered();
}
} //clearing string for next read
}

//////////////////// Functions //////////////////////////////////////

//take care of some special events
void keypadEvent(KeypadEvent eKey){
    switch (keypad.getState()){
        case PRESSED:
            if (passwd_pos - 12 >= 5) {
                return ;
            }
            lcd.setCursor((passwd_pos++),0);
            switch (eKey){
                case '#':          // # is to validate password
                    passwd_pos = 12;
                    checkPassword();

                    break;
                case '*':          // * is to reset password attempt
                    password.reset();
                    passwd_pos = 12;
            }
        }
    }
}

```

```
// TODO: clear the screen output

    break;

default:

    password.append(eKey);

    lcd.print("*");

}

}

}

void alarmTriggered(){

    int expected_pos;

    int incr;

    digitalWrite(speakerPin, HIGH);

    digitalWrite(reedPin1, HIGH);

    digitalWrite(relay2, LOW);

    digitalWrite(relay3, LOW);

    //

    password.reset();

    alarmStatus = 0;

    //alarmActive = 1;

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("SYSTEM TRIGGERED");

    //client.publish("bbb", "SYSTEM TRIGGERED");

    lcd.setCursor(0,4);

    if (zone == 1)

    {

        lcd.print("Front Door Open");

        client.publish("bbb", "Front Door Open");
```

```
    expected_pos = 65;
    delay(200);
}

if(zone == 0){
    expected_pos = 40;
    lcd.print("Motion in Bedroom");
    client.publish("bbb", "Motion in Bedroom");
    delay(500);
}

else if(zone == 2){
    expected_pos = 10;
    lcd.print("Backdoor Open");
    client.publish("bbb", "Backdoor Open");
    delay(500);
}

else if(zone == 3){
    expected_pos = 145;
    lcd.print("Motion in Server");
    client.publish("bbb", "Motion in Server");
    delay(500);
}

digitalWrite(speakerPin, LOW);
digitalWrite(reedPin1, HIGH);

if (!client.connected()) {
    reconnect();
} else {
    client.publish("bbb", "");
}
```

```
    expected_pos = 65;
    delay(200);
}

if(zone == 0){
    expected_pos = 40;
    lcd.print("Motion in Bedroom");
    client.publish("bbb", "Motion in Bedroom");
    delay(500);
}

else if(zone == 2){
    expected_pos = 10;
    lcd.print("Backdoor Open");
    client.publish("bbb", "Backdoor Open");
    delay(500);
}

else if(zone == 3){
    expected_pos = 145;
    lcd.print("Motion in Server");
    client.publish("bbb", "Motion in Server");
    delay(500);
}

digitalWrite(speakerPin, LOW);
digitalWrite(reedPin1, HIGH);

if (!client.connected()) {
    reconnect();
} else {
    client.publish("bbb", "");
}
```

```

}

}

void checkPassword(){           // To check if PIN is corrected, if not, retry!
    if (password.evaluate())
    {
        if(alarmActive == 0 && alarmStatus == 0)
        {
            activate();
        }
        else if( alarmActive == 1 || alarmStatus == 1) {
            deactivate();
        }
    }
    else {
        invalidCode();
    }
}

void invalidCode() // display meaasge when a invalid is entered
{
    password.reset();
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("INVALID CODE! LOL!");
    lcd.setCursor(5,2);
    lcd.print("TRY AGAIN!");
    digitalWrite(greenLED, LOW);
    digitalWrite(redLED, HIGH);
}

```



```
    delay(2000);
    digitalWrite(redLED, LOW);
    delay(1000);
    displayCodeEntryScreen();
}

void activate()    // Activate the system if correct PIN entered and display message on the screen
{
    if (digitalRead(reedPin1) == HIGH){
        digitalWrite(speakerPin, HIGH);
    } else {
        digitalWrite(speakerPin, LOW);
    }
    digitalWrite(reedPin2, HIGH);
    digitalWrite(relay2, LOW);
    digitalWrite(redLED, HIGH);
    digitalWrite(greenLED, LOW);
    digitalWrite(2, HIGH);
    lcd.setCursor(0,0);
    lcd.print("SYSTEM ACTIVE!");
    alarmActive = 1;
    password.reset();
    delay(2000);
}

void deactivate()
{
    //digitalWrite(camera, LOW);
    alarmStatus = 0;
```

```
digitalWrite(reedPin1, LOW);
digitalWrite(reedPin2, LOW);
digitalWrite(redLED, LOW);
digitalWrite(greenLED, HIGH);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("SYSTEM DEACTIVATED!");
digitalWrite(speakerPin, LOW);
alarmActive = 0;
password.reset();
delay(5000);
digitalWrite(relay2, HIGH);
digitalWrite(relay3, HIGH);
displayCodeEntryScreen();
}
void displayCodeEntryScreen() // Displaying start screen for users to enter PIN
{
  lcd.clear();
  lcd.setBacklight(HIGH);
  lcd.setCursor(0,0);
  lcd.print("Enter PIN:");
  lcd.setBacklight(HIGH);
}
```

Uji Coba

