

## CURRICULUM VITAE

NAMA : YURIS RAMADHONA  
NIM : 0614 4035 1544  
TEMPAT, TANGGAL LAHIR : PALEMBANG, 30 JANUARI 1997  
ALAMAT : JL. PAHLAWAN KEMARUNG  
Gg.DURIAN,BATURAJA  
PHONE : 087796243565

### PENDIDIKAN FORMAL :

PENDIDIKAN	NAMA SEKOLAH	TAHUN LULUS
SEKOLAH DASAR	SDN 49 OKU BATURAJA	2009
SMP	SMPN 1 OKU BATURAJA	2012
SMA	SMAN 5 OKU BATURAJA	2015

### PENDIDIKAN NON-FORMAL :

PENDIDIKAN	TAHUN
FANDO COURSE	2009

### PENGALAMAN PENELITIAN :

No	PENELITIAN	TAHUN
1	PERANCANGAN ALAT PENDETEKSI KEBOCORAN <i>LIQUFIED PETROLIUM GAS</i> (LPG) BERBASIS IOT	2018

### PENGHARGAAN/PENCAPAIAN :

No	PENGHARGAAN	TAHUN
1	SEMINAR ETIKA MOTIASI DAN SEMINAR ETIKA ENGINEER PREN	2016
2	PANITIA HUMAS DAN DOKUMENTASI PELATIHAN PKM	2017
3	SEMINAR KEPEMIMPINAN	2017
4	PENARI KOLOSAL DIES NATALIS 35TH	2018
4	SEMINAR KOLABORASI BAWASLU DAN IKA-UNSRI	2018
5	PANITIA KEGIATAN HIPMI EXPO	2018

### PENGALAMAN ORGANISASI :

No	NAMA ORGANISASI	TAHUN
1.	SEKRETARIS FORUM SILATURAHMI MAHASISWA BATURAJA (FOSMAB) DI PALEMBANG	2015
2.	BENDAHARA FORUM SILATURAHMI MAHASISWA BATURAJA (FOSMAB) DI PALEMBANG	2016

3.	ANGGOTA AKTIF DEPARTEMENT <i>SAVING ENERGY CAMPUS (SECA)</i> UKM MARS POLSRI	2016
4.	ANGGOTA AKTIF DEPARTEMENT RISTEK UKM MARS POLSRI	2017
5.	MAGANG DI PT. Industri Telekomunikasi Indonesia (persero) BANDUNG	2018

Semua data yang saya tulis dan isi di Curriculum Vitae ini benar dan dapat dipertanggung jawabkan.

**PALEMBANG, Juli 2019**

**(Yuris Ramadhona)**

## PENGANGGARAN BIAYA

No.	Material	Justifikasi Pemakaian	Kuantitas	Harga Satuan (Rp)	Harga Peralatan Penunjang (Rp)
<b>A. Peralatan Penunjang</b>					
1.	Komponen yang dibutuhkan	Alat Pembuatan Perangkat Keras	1 set	1.965.000	1.965.000
2.	Software	Alat Pembuatan Perangkat Lunak	1 buah	50.000	50.000
<b>B. Lain-lain</b>					
1.	Pembuatan Buku Proposal	Alat Persyaratan Pra TA	5 buah	30.000	150.000
2.	Pembuatan Tugas Akhir	Alat Persyaratan TA	6 buah	50.000	300.000
3.	Publikasi, Seminar dll	Alat Persyaratan TA	1 kali	350.000	350.000
4.	Kertas A4 80gr	Alat Pembuatan Proposal dan Laporan	4 rim	50.000	200.000
5	ATK	Penunjang	1 set	400.00	400.000
<b>TOTAL</b>					3.415.000

### RINCIAN HARGA KOMPONEN

No.	Material	Jumlah	Harga Satuan (Rp)	Total Harga (Rp)
1.	Raspberri Pi 3 model B	1	900.000	900.000
2.	Modul GPS	1	200.000	200.000
3.	Sensor PIR	1	30.000	30.000
4.	Sensor Gas MQ-6	1	50.000	50.000
5.	Sensor DHT11	1	40.000	40.000
6.	Relay 4 channel	2	90.000	180.000
7.	Power Supply	1	150.000	150.000
8.	Buzzer	1	10.000	10.000
9.	Lampu 3 Watt	4	30.000	120.000
11.	Kipas Raspi	1	35.000	35.000
12.	Biaya Lain-lain		700.000	700.000
Total Biaya			Rp. 1.965.000	

## List Program Netbeans

```
#include <wiringPi.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <inttypes.h> // uint8_t, etc
#include <linux/i2c-dev.h> // I2C bus definitions
#include <sys/ioctl.h>
#include "wiringPi.h"
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <wiringSerial.h>
#include <stdlib.h>
#include <stdint.h>
#include <time.h>
#include <iostream>
#include <vector>
#include <curl/curl.h>
#include <linux/i2c-dev.h>
#include <linux/i2c.h>
#include <sys/ioctl.h>
#include <math.h>
#include <stdint.h>
#include <string.h>
#define MAXTIMINGS 85
#define DHTPIN 7

using namespace std;
string Tingkat_Kebocoran_Gas, Status_Kebocoran, Lampu_1, Lampu_2, Lampu_3, Lampu_4, Status_PIR;
string Lampu_1_sebelumnya = "";
string Lampu_2_sebelumnya = "";
string Lampu_3_sebelumnya = "";
string s_gas;
string s_suhu;
float Suhu;
double d_lat, d_long;

//fuzzy
double tidak_bocor = 0;
double bocor_sedikit = 0;
double bocor_banyak = 0;
double berbahaya = 0;
double suhu_normal = 0;
double suhu_hangat = 0;
double suhu_panas = 0;
double keadaan_aman = 50;
double keadaan_tidak_aman = 100;
double keadaan_bahaya = 200;
double wz, w, data_keluaran;

//koding ads
int fd;
int asd_address = 0x48;
int16_t val;
uint8_t writeBuf[3];
uint8_t readBuf[2];
float myfloat;
```

```
const float VPS = 4.096 / 32768.0; // volts per step
int dht11_dat[5] = {0, 0, 0, 0, 0};
```

```
void kirim_server();
```

```
struct MemoryStruct {
    char *memory;
    size_t size;
};
static size_t
WriteMemoryCallback(void *contents, size_t size, size_t nmemb, void *userp) {
    size_t realsize = size * nmemb;
    struct MemoryStruct *mem = (struct MemoryStruct *) userp;

    mem->memory = (char *) realloc(mem->memory, mem->size + realsize + 1);
    if (mem->memory == NULL) {
        /* out of memory! */
        printf("not enough memory (realloc returned NULL)\n");
        return 0;
    }

    memcpy(&(mem->memory[mem->size]), contents, realsize);
    mem->size += realsize;
    mem->memory[mem->size] = 0;

    return realsize;
}

vector<string> split(string s, string delimiter) {
    size_t pos_start = 0, pos_end, delim_len = delimiter.length();
    string token;
    vector<string> res;
    while ((pos_end = s.find(delimiter, pos_start)) != string::npos) {
        token = s.substr(pos_start, pos_end - pos_start);
        pos_start = pos_end + delim_len;
        res.push_back(token);
    }

    res.push_back(s.substr(pos_start));
    return res;
}

void read_dht11_dat() {
    uint8_t laststate = HIGH;
    uint8_t counter = 0;
    uint8_t j = 0, i;
    float f; /* fahrenheit */

    struct timespec tim, tim2;
    //    tim.tv_sec = 1;
    tim.tv_nsec = 1500;
    dht11_dat[0] = dht11_dat[1] = dht11_dat[2] = dht11_dat[3] = dht11_dat[4] = 0;

    /* pull pin down for 18 milliseconds */
    pinMode(DHTPIN, OUTPUT);
    digitalWrite(DHTPIN, LOW);
    delay(18);
    /* then pull it up for 40 microseconds */
    digitalWrite(DHTPIN, HIGH);
    delayMicroseconds(40);
```

```

/* prepare to read the pin */
pinMode(DHTPIN, INPUT);
/* detect change and read data */
for (i = 0; i < MAXTIMINGS; i++) {
    counter = 0;
    while (digitalRead(DHTPIN) == laststate) {
        counter++;
// nanosleep(&tim , &tim2);
        delayMicroseconds(3);
// delayNanoseconds(1500);
// nanosleep(1500);
// if(nanosleep(&tim , &tim2) < 0 )
// {
//     printf("Nano sleep system call failed \n");
//     return -1;
// }
        if (counter == 255) {
            break;
        }
    }
    laststate = digitalRead(DHTPIN);

    if (counter == 255)
        break;

    /* ignore first 3 transitions */
    if ((i >= 4) && (i % 2 == 0)) {
        /* shove each bit into the storage bytes */
        dht11_dat[j / 8] <<= 1;
        if (counter > 16)
            dht11_dat[j / 8] |= 1;
        j++;
    }
}

/*
* check we read 40 bits (8bit x 5 ) + verify checksum in the last byte
* print it out if data is good
*/
if ((j >= 40) &&
    (dht11_dat[4] == ((dht11_dat[0] + dht11_dat[1] + dht11_dat[2] + dht11_dat[3]) & 0xFF))) {
    f = dht11_dat[2] * 9. / 5. + 32;
    printf("Humidity = %d.%d %% Temperature = %d.%d *C (%.1f *F)\n",
        dht11_dat[0], dht11_dat[1], dht11_dat[2], dht11_dat[3], f);
    Suhu = dht11_dat[2];
    printf("Data not good, skip\n");
}
}

```

```

void defuzzyfikasi(double dk1, double dk2, int kondisi) {
    double dk_terendah = 1;
    if (dk1 < dk_terendah) dk_terendah = dk1;
    if (dk2 < dk_terendah) dk_terendah = dk2;

    wz += dk_terendah * kondisi;
    w += dk_terendah;
}

```

```

double derajat_keanggotaan_trapesium(double Gas, int titik1, int titik2, int titik3, int titik4) {
    double dk_medium = 0;

```

```

if (Gas <= titik1) {
    dk_medium = 0;
} else if (Gas <= titik2) {
    dk_medium = (Gas - titik1) / (titik2 - titik1);
} else if (Gas <= titik3) {
    dk_medium = 1;
} else if (Gas <= titik4) {
    dk_medium = (titik4 - Gas) / (titik4 - titik3);
} else {
    dk_medium = 0;
}
return dk_medium;
}

```

```

void hitung_derajat_keanggotaan(double Gas, double Suhu) {
    w = 0;
    wz = 0;

```

//Fuzzy MQ6

```

if (Gas < 0) Gas = 0 - Gas;
tidak_bocor = derajat_keanggotaan_trapesium(Gas, -1, -1, 1, 1.5);
bocor_sedikit = derajat_keanggotaan_trapesium(Gas, 1, 1.5, 2, 2.5);
bocor_banyak = derajat_keanggotaan_trapesium(Gas, 2, 2.5, 3, 3.5);
berbahaya = derajat_keanggotaan_trapesium(Gas, 3, 3.5, 4.5, 0);

if (tidak_bocor > bocor_sedikit && tidak_bocor > bocor_banyak && tidak_bocor > berbahaya) {
    s_gas = "Tidak Bocor";
}

if (bocor_sedikit > tidak_bocor && bocor_sedikit > bocor_banyak && bocor_sedikit > berbahaya) {
    s_gas = "Bocor Sedikit";
}

if (bocor_banyak > tidak_bocor && bocor_banyak > bocor_sedikit && bocor_banyak > berbahaya) {
    s_gas = "Bocor Banyak";
}

if (berbahaya > tidak_bocor && berbahaya > bocor_sedikit && berbahaya > bocor_banyak) {
    s_gas = "Berbahaya";
}

```

//Fuzzy DHT11

```

suhu_normal = derajat_keanggotaan_trapesium(Suhu, -1, -1, 21, 25);
suhu_hangat = derajat_keanggotaan_trapesium(Suhu, 21, 30, 35, 40);
suhu_panas = derajat_keanggotaan_trapesium(Suhu, 30, 40, 50, 0);

if (suhu_normal > suhu_hangat && suhu_normal > suhu_panas) {
    s_suhu = "Suhu Normal";
}

if (suhu_hangat > suhu_normal && suhu_hangat > suhu_panas) {
    s_suhu = "Suhu Hangat";
}

if (suhu_panas > suhu_normal && suhu_panas > suhu_hangat) {
    s_suhu = "Suhu Panas";
}

```



```

//Fuzzy
cout<< tidak_bocor << " " << bocor_sedikit << " " << bocor_banyak << " " << berbahaya << " " <<
suhu_normal << " " << suhu_hangat << " " << suhu_panas << endl;

```

```

}

```

```

int rule_base1() {
    w = 0;

    if (tidak_bocor > 0 && suhu_normal > 0) {
        defuzzyfikasi(tidak_bocor, suhu_normal, keadaan_aman);
    }
    if (tidak_bocor > 0 && suhu_hangat > 0) {
        defuzzyfikasi(tidak_bocor, suhu_hangat, keadaan_aman);
    }
    if (tidak_bocor > 0 && suhu_panas > 0) {
        defuzzyfikasi(tidak_bocor, suhu_panas, keadaan_aman);
    }
    if (bocor_sedikit > 0 && suhu_normal > 0) {
        defuzzyfikasi(bocor_sedikit, suhu_normal, keadaan_tidak_aman);
    }
    if (bocor_sedikit > 0 && suhu_hangat > 0) {
        defuzzyfikasi(bocor_sedikit, suhu_hangat, keadaan_tidak_aman);
    }
    if (bocor_sedikit > 0 && suhu_panas > 0) {
        defuzzyfikasi(bocor_sedikit, suhu_panas, keadaan_bahaya);
    }
    if (bocor_banyak > 0 && suhu_normal > 0) {
        defuzzyfikasi(bocor_banyak, suhu_normal, keadaan_tidak_aman);
    }
    if (bocor_banyak > 0 && suhu_hangat > 0) {
        defuzzyfikasi(bocor_banyak, suhu_hangat, keadaan_bahaya);
    }
    if (bocor_banyak > 0 && suhu_panas > 0) {
        defuzzyfikasi(bocor_banyak, suhu_panas, keadaan_bahaya);
    }
    if (berbahaya > 0 && suhu_normal > 0) {
        defuzzyfikasi(berbahaya, suhu_normal, keadaan_bahaya);
    }
    if (berbahaya > 0 && suhu_hangat > 0) {
        defuzzyfikasi(berbahaya, suhu_hangat, keadaan_bahaya);
    }
    if (berbahaya > 0 && suhu_panas > 0) {
        defuzzyfikasi(berbahaya, suhu_panas, keadaan_bahaya);
    }
}

data_keluaran = wz / w;
cout << "Hasil Fuzzy : " << data_keluaran << " " << w << " " << wz << endl;

```

```

//Capture Hasil Fuzzy

```

```

if (data_keluaran < 100) {
    Status_Kebocoran = "Aman";
    cout << "Aman" << endl;
} else if (data_keluaran < 150) {
    Status_Kebocoran = "Tidak Aman";
    cout << "Tidak Aman" << endl;
} else {
    Status_Kebocoran = "Bahaya";
    cout << "Bahaya" << endl;
}

```

```

//Capture Hasil Fuzzy
return (int) data_keluaran;
}

void kirim_data() {
    CURL *curl;
    CURLcode res;

    struct MemoryStruct chunk;

    chunk.memory = (char*) malloc(1); /* will be grown as needed by the realloc above */
    chunk.size = 0; /* no data at this point */

    struct curl_httppost *formpost = NULL;
    struct curl_httppost *lastptr = NULL;
    struct curl_slist *headerlist = NULL;
    static const char buff[] = "Expect:";

    curl_global_init(CURL_GLOBAL_ALL);
    /* Fill in the file upload field */
    char buff_Lampu_1[Lampu_1.size() + 1];
    strcpy(buff_Lampu_1, Lampu_1.c_str());

    char buff_Lampu_2[Lampu_2.size() + 1];
    strcpy(buff_Lampu_2, Lampu_2.c_str());

    char buff_Lampu_3[Lampu_3.size() + 1];
    strcpy(buff_Lampu_3, Lampu_3.c_str());

    curl_formadd(&formpost,
                &lastptr,
                CURLFORM_COPYNAME, "lampu_1",
                CURLFORM_COPYCONTENTS, buff_Lampu_1,
                CURLFORM_END);
    curl_formadd(&formpost,
                &lastptr,
                CURLFORM_COPYNAME, "lampu_2",
                CURLFORM_COPYCONTENTS, buff_Lampu_2,
                CURLFORM_END);
    curl_formadd(&formpost,
                &lastptr,
                CURLFORM_COPYNAME, "lampu_3",
                CURLFORM_COPYCONTENTS, buff_Lampu_3,
                CURLFORM_END);

    /* Fill in the submit field too, even if this is rarely needed */
    curl_formadd(&formpost,
                &lastptr,
                CURLFORM_COPYNAME, "submit",
                CURLFORM_COPYCONTENTS, "send",
                CURLFORM_END);

    curl = curl_easy_init();

    if (curl) {

        curl_easy_setopt(curl, CURLOPT_SSL_VERIFYPEER, false);

        /* send all data to this function */
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, WriteMemoryCallback);
    }
}

```

```

/* we pass our 'chunk' struct to the callback function */
curl_easy_setopt(curl, CURLOPT_WRITEDATA, (void *) &chunk);

/* some servers don't like requests that are made without a user-agent
field, so we provide curl */
curl_easy_setopt(curl, CURLOPT_USERAGENT, "my3dsoftware");
/* initialize custom header list (stating that Expect: 100-continue is not
wanted */
headerlist = curl_slist_append(headerlist, buf);

/* what URL that receives this POST */
//curl_easy_setopt(curl, CURLOPT_URL, "http://api.sketchfab.com/v1/models");
curl_easy_setopt(curl, CURLOPT_URL,
"http://telkom15.com/YurisRamadhona/raspi/uploadpengendali.php");

// if ((argc == 2) && (!strcmp(argv[1], "noexpectheader")))
// /* only disable 100-continue header if explicitly requested */
// curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headerlist);
curl_easy_setopt(curl, CURLOPT_HTTPPOST, formpost);

/* Perform the request, res will get the return code */
res = curl_easy_perform(curl);
/* Check for errors */
if (res != CURLE_OK) {
    fprintf(stderr, "curl_easy_perform() failed: %s\n",
        curl_easy_strerror(res));
} else {
    // printf("%lu bytes retrieved\n", (long)chunk.size);
    // printf("hasil : %s\n", chunk.memory);
    char buff_data_masuk[200];
    sprintf(buff_data_masuk, "%s", chunk.memory);
    string pesan_masuk(buff_data_masuk);
    cout << pesan_masuk << endl;
    if (pesan_masuk.find("\Lampu_1\":"DIHIDUPKAN") != string::npos) {
        cout << "lampu 1 hidup" << endl;
        digitalWrite(4, LOW);
        Lampu_1 = "ON";
    } else {
        cout << "lampu 1 mati" << endl;
        digitalWrite(4, HIGH);
        Lampu_1 = "OFF";
    }
    if (pesan_masuk.find("\Lampu_2\":"DIHIDUPKAN") != string::npos) {
        cout << "lampu 2 hidup" << endl;
        digitalWrite(5, LOW);
        Lampu_2 = "ON";
    } else {
        cout << "lampu 2 mati" << endl;
        digitalWrite(5, HIGH);
        Lampu_2 = "OFF";
    }
    if (pesan_masuk.find("\Lampu_3\":"DIHIDUPKAN") != string::npos) {
        cout << "lampu 3 hidup" << endl;
        digitalWrite(6, LOW);
        Lampu_3 = "ON";
    } else {
        cout << "lampu 3 mati" << endl;
        digitalWrite(6, HIGH);
        Lampu_3 = "OFF";
    }
}
}

```

```

    }
}

/* always cleanup */
curl_easy_cleanup(curl);

/* then cleanup the formpost chain */
curl_formfree(formpost);
/* free slist */
curl_slist_free_all(headerlist);
}

void kirim_sensorpir() {

    CURL *curl;
    CURLcode res;

    struct MemoryStruct chunk;

    chunk.memory = (char*) malloc(1); /* will be grown as needed by the realloc above */
    chunk.size = 0; /* no data at this point */
    struct curl_httppost *formpost = NULL;
    struct curl_httppost *lastptr = NULL;
    struct curl_slist *headerlist = NULL;
    static const char buf[] = "Expect:";

    curl_global_init(CURL_GLOBAL_ALL);

    /* Fill in the file upload field */
    char buff_Status_PIR[Status_PIR.size() + 1];
    strcpy(buff_Status_PIR, Status_PIR.c_str());
    cout << Status_PIR << endl;
    printf("%s", buff_Status_PIR);
    char buff_Suhu[50] = { };
    sprintf(buff_Suhu, "%.0f", Suhu);

    curl_formadd(&formpost,
                &lastptr,
                CURLFORM_COPYNAME, "status_pir",
                CURLFORM_COPYCONTENTS, buff_Status_PIR,
                CURLFORM_END);

    curl_formadd(&formpost,
                &lastptr,
                CURLFORM_COPYNAME, "Suhu",
                CURLFORM_COPYCONTENTS, buff_Suhu,
                CURLFORM_END);

    /* Fill in the submit field too, even if this is rarely needed */
    curl_formadd(&formpost,
                &lastptr,
                CURLFORM_COPYNAME, "submit",
                CURLFORM_COPYCONTENTS, "send",
                CURLFORM_END);

    curl = curl_easy_init();

    if (curl) {

        curl_easy_setopt(curl, CURLOPT_SSL_VERIFYPEER, false);

```

```

/* send all data to this function */
curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, WriteMemoryCallback);

/* we pass our 'chunk' struct to the callback function */
curl_easy_setopt(curl, CURLOPT_WRITEDATA, (void *) &chunk);

/* some servers don't like requests that are made without a user-agent
field, so we provide curl */
curl_easy_setopt(curl, CURLOPT_USERAGENT, "my3dsoftware");

/* initialize custom header list (stating that Expect: 100-continue is not
wanted */
headerlist = curl_slist_append(headerlist, buf);

/* what URL that receives this POST */
//curl_easy_setopt(curl, CURLOPT_URL, "http://api.sketchfab.com/v1/models");
curl_easy_setopt(curl, CURLOPT_URL, "http://telkom15.com/YurisRamadhona/raspi/uploadpiroke.php");
// if ((argc == 2) && (!strcmp(argv[1], "noexpectheader")))
/* only disable 100-continue header if explicitly requested */
curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headerlist);
curl_easy_setopt(curl, CURLOPT_HTTPPOST, formpost);

/* Perform the request, res will get the return code */
res = curl_easy_perform(curl);
/* Check for errors */
if (res != CURLE_OK) {
    fprintf(stderr, "curl_easy_perform() failed: %s\n",
        curl_easy_strerror(res));
} else {
// printf("%lu bytes retrieved\n", (long)chunk.size);
    printf("%s\n", chunk.memory);
}

/* always cleanup */
curl_easy_cleanup(curl);

/* then cleanup the formpost chain */
curl_formfree(formpost);
/* free slist */
curl_slist_free_all(headerlist);
}
}

```

```
void kirim_datagas() {
```

```

    CURL *curl;
    CURLcode res;

```

```

    struct MemoryStruct chunk;

```

```

    chunk.memory = (char*) malloc(1); /* will be grown as needed by the realloc above */
    chunk.size = 0; /* no data at this point */
    struct curl_httppost *formpost = NULL;
    struct curl_httppost *lastptr = NULL;
    struct curl_slist *headerlist = NULL;
    static const char buf[] = "Expect:";

```

```

    curl_global_init(CURL_GLOBAL_ALL);

```

```

/* Fill in the file upload field */
char buff_Tingkat_Kebocoran_Gas[Tingkat_Kebocoran_Gas.size() + 1];
strcpy(buff_Tingkat_Kebocoran_Gas, Tingkat_Kebocoran_Gas.c_str());
cout << Tingkat_Kebocoran_Gas << endl;

char buff_Status_Kebocoran[Status_Kebocoran.size() + 1];
strcpy(buff_Status_Kebocoran, Status_Kebocoran.c_str());
cout << Status_Kebocoran << endl;

char buff_lat[50] = { };
sprintf(buff_lat, "%4.6f", d_lat);

char buff_long[50] = { };
sprintf(buff_long, "%4.6f", d_long);

curl_formadd(&formpost,
             &lastptr,
             CURLFORM_COPYNAME, "Tingkat_Kebocoran_Gas",
             CURLFORM_COPYCONTENTS, buff_Tingkat_Kebocoran_Gas,
             CURLFORM_END);
curl_formadd(&formpost,
             &lastptr,
             CURLFORM_COPYNAME, "Status_Kebocoran",
             CURLFORM_COPYCONTENTS, buff_Status_Kebocoran,
             CURLFORM_END);
curl_formadd(&formpost,
             &lastptr,
             CURLFORM_COPYNAME, "Latitude",
             CURLFORM_COPYCONTENTS, buff_lat,
             CURLFORM_END);
curl_formadd(&formpost,
             &lastptr,
             CURLFORM_COPYNAME, "Longitude",
             CURLFORM_COPYCONTENTS, buff_long,
             CURLFORM_END);

/* Fill in the submit field too, even if this is rarely needed */
curl_formadd(&formpost,
             &lastptr,
             CURLFORM_COPYNAME, "submit",
             CURLFORM_COPYCONTENTS, "send",
             CURLFORM_END);

curl = curl_easy_init();

if (curl) {

    curl_easy_setopt(curl, CURLOPT_SSL_VERIFYPEER, false);

    /* send all data to this function */
    curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, WriteMemoryCallback);

    /* we pass our 'chunk' struct to the callback function */
    curl_easy_setopt(curl, CURLOPT_WRITEDATA, (void *) &chunk);

    /* some servers don't like requests that are made without a user-agent
    field, so we provide curl */
    curl_easy_setopt(curl, CURLOPT_USERAGENT, "my3dsoftware");

    /* initialize custom header list (stating that Expect: 100-continue is not

```

```

wanted */
headerlist = curl_slist_append(headerlist, buf);

/* what URL that receives this POST */
/*curl_easy_setopt(curl, CURLOPT_URL, "http://api.sketchfab.com/v1/models");
curl_easy_setopt(curl, CURLOPT_URL, "http://telkom15.com/YurisRamadhona/raspi/uploadgas.php");

// if ((argc == 2) && (!strcmp(argv[1], "noexpectheader")))
/* only disable 100-continue header if explicitly requested */
curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headerlist);
curl_easy_setopt(curl, CURLOPT_HTTPPOST, formpost);

/* Perform the request, res will get the return code */
res = curl_easy_perform(curl);
/* Check for errors */
if (res != CURLE_OK) {
    fprintf(stderr, "curl_easy_perform() failed: %s\n",
            curl_easy_strerror(res));
} else {
    // printf("%lu bytes retrieved\n", (long)chunk.size);
    printf("%s\n", chunk.memory);
}

/* always cleanup */
curl_easy_cleanup(curl);

/* then cleanup the formpost chain */
curl_formfree(formpost);
/* free slist */
curl_slist_free_all(headerlist);
}
}

void hidup_Lampu_1() {
    digitalWrite(4, LOW);
    Lampu_1 = "ON";
}

void mati_Lampu_1() {
    digitalWrite(4, HIGH);
    Lampu_1 = "OFF";
}

void hidup_Lampu_2() {
    digitalWrite(5, LOW);
    Lampu_2 = "ON";
}

void mati_Lampu_2() {
    digitalWrite(5, HIGH);
    Lampu_2 = "OFF";
}

void hidup_Lampu_3() {
    digitalWrite(6, LOW);
    Lampu_3 = "ON";
}

void mati_Lampu_3() {
    digitalWrite(6, HIGH);
}

```

```

    Lampu_3 = "OFF";
}

void hidup_Lampu_4() {
    digitalWrite(29, LOW);
    Lampu_4 = "ON";
}

void mati_Lampu_4() {
    digitalWrite(29, HIGH);
    Lampu_4 = "OFF";
}

int main(void) {

    Lampu_1 = "Off";
    Lampu_2 = "Off";
    Lampu_3 = "Off";

    printf("Raspberry Pi wiringPi DHT11 Temperature test program\n");

    if (wiringPiSetup() == -1)
        exit(1);

//pir
    if (wiringPiSetup() == -1) {
        printf("%s \n", "Failed to setup wiringPi");
    }

    pinMode(1, INPUT); // set pin 1 as input
    pullUpDnControl(1, PUD_UP);

//koding Relay
    pinMode(4, OUTPUT); // set pin 16 as input
    pinMode(5, OUTPUT); // set pin 18 as input
    pinMode(6, OUTPUT); // set pin 22 as input
    pinMode(29, OUTPUT); // set pin 40 as input

//koding buzzer
    pinMode(28, OUTPUT); // set pin 38 as input

//koding ads
    if ((fd = open("/dev/i2c-1", O_RDWR)) < 0) {
        printf("Error: Couldn't open device! %d\n", fd);
        exit(1);
    }

// connect to ADS1115 as i2c slave
    if (ioctl(fd, I2C_SLAVE, asd_address) < 0) {
        printf("Error: Couldn't find device on address!\n");
        exit(1);
    }
    writeBuf[0] = 1; // config register is 1
    writeBuf[1] = 0b11000010; // 0xC2 single shot off
    writeBuf[2] = 0b10000101;
    if (write(fd, writeBuf, 3) != 3) {
        perror("Write to register 1");
        exit(1);
    }
}

```



```

//sleep(1);

printf("ASD1115 Demo will take five readings.\n");

// set pointer to 0
readBuf[0] = 0;
if (write(fd, readBuf, 1) != 1) {
    perror("Write register select");
    exit(-1);
}
// take 5 readings:

int count = 1;

//gps
int serial_port;
char dat, buff[100], GGA_code[3];
unsigned char IsitGGAstring = 0;
unsigned char GGA_index = 0;
unsigned char is_GGA_received_completely = 0;

if ((serial_port = serialOpen("/dev/ttyS0", 9600)) < 0) /* open serial port */ {
    fprintf(stderr, "Unable to open serial device: %s\n", strerror(errno));
    return 1;
}

if (wiringPiSetup() == -1) /* initializes wiringPi setup */ {
    fprintf(stdout, "Unable to start wiringPi: %s\n", strerror(errno));
    return 1;
}

while (1) {
    try {
        read_dht11_dat();
        delay(1000); /* wait 1sec to refresh */

        //gps
        while (serialDataAvail(serial_port)) /* check for any data available on serial port */ {
            dat = serialGetchar(serial_port); /* receive character serially */
            if (dat == '\r') {
                IsitGGAstring = 0;
                GGA_index = 0;
                is_GGA_received_completely = 1;
                // printf("GGA: %s",buff);
            } else {
                buff[GGA_index++] = dat;
                //
            }
        }
        if (is_GGA_received_completely == 1) {
            // printf("GGA: %s",buff);
            string pesan(buff);
            cout << pesan << endl;
            // try {
            //string str = "adsf+qwret+nvfkbdsj-+orthdfjgh-+dfjrleih";
            string delimiter = "$GPGLL,";
            //split berdasarkan GPGLL
            vector<string> v = split(pesan, delimiter);
            cout << "sise:" << v.size() << endl;
            cout << "sise:" << v[0] << endl;
        }
    }
}

```

```

if (v.size() > 1) {
    cout << v[1] << endl;
    //prosesambilnilai latdan long

    //pecah gppllberdasarkan koma
    vector<string> w = split(v[1], ",");
    string latitude = w[0];
    string longitude = w[2];
    vector<string> latt = split(latitude, ".");
    vector<string> longg = split(longitude, ".");

    cout << w[0] << " " << w[1] << w[2] << endl;
    if (latitude.length() > 0) {
        d_lat = 0 - (int) (stof(latt[0]) / 100) - (stof(latt[0].substr(2, 2)) / 60) - (stof(latt[1]) / 36000000);
        d_long = (int) (stof(longg[0]) / 100) + (stof(longg[0].substr(3, 2)) / 60) + (stof(longg[1]) /
36000000);
        //          d_lat = 0 - stof(latitude) / 100;
        //          d_long = stof(longitude) / 100;
        printf("float: %4.6f %4.6f ", d_lat, d_long);
        //          sleep(1);

        //          cout << d_lat << " " << d_long << endl;
        //          cout << latitude << " " << longitude << endl;
    }

    //      } catch (int e){
    //
    //      }
    //      for (auto i : v) cout << i << endl;

    //
    //          vector<string> pecahan = split(pesan,"GPPLL");
    //          if (pecahan.size() > 0){
    //              cout<<pecahan[1]<<endl;
    //          }
    is_GGA_received_completely = 0;

}

// pir
int val = digitalRead(1);
printf("%s: %i\n", "sensor", val);
//usleep(100000); // 10fps
if (val == 1) {

// Koding Ralay
    digitalWrite(29, LOW);
    //    digitalWrite(6, LOW);
    //    digitalWrite(5, LOW);
    //    digitalWrite(4, LOW);
    Status_PIR = "Terdeteksi Manusia";
} else { // Koding Ralay
    digitalWrite(29, HIGH);
    //    digitalWrite(6, HIGH);
    //    digitalWrite(5, HIGH);
    //    digitalWrite(4, HIGH);
    Status_PIR = "Tidak Terdeteksi Manusia";
}
if (Lampu_1 == Lampu_1_sebelumnya && Lampu_2 == Lampu_2_sebelumnya && Lampu_3 ==
Lampu_3_sebelumnya) {

```

```

} else {
    Lampu_1_sebelumnya = Lampu_1;
    Lampu_2_sebelumnya = Lampu_2;
    Lampu_3_sebelumnya = Lampu_3;
}

```

**//koding ads**

```

if (read(fd, readBuf, 2) != 2) {
    perror("Read conversion");
    // exit(-1);
} else {

    // could also multiply by 256 then add readBuf[1]
    val = readBuf[0] << 8 | readBuf[1];

    // with +- LSB sometimes generates very low neg number.
    if (val < 0) val = 0;

    myfloat = val * VPS; // convert to voltage

    printf("Conversion number %d HEX 0x%02x DEC %d %4.3f volts.\n",
        count, val, val, myfloat);
    // TMP37 20mV per deg C
    printf("Temp. C = %4.2f\n", myfloat / 0.02);

    //printf("Temp. F = %4.2f\n", myfloat / 0.02 * 9 / 5 + 32);
    if (myfloat > 2) {
        digitalWrite(28, HIGH);
        printf("Bunyi \n");
        Tingkat_Kebocoran_Gas = to_string(myfloat);
    } else {
        digitalWrite(28, LOW);
        printf("Tidak Bunyi \n");
        Tingkat_Kebocoran_Gas = to_string(myfloat);
    }
    // sleep(5);
}

```

```

cout <<myfloat<<" "<<Suhu<<endl;
hitung_derajat_keanggotaan(myfloat, Suhu);
rule_base1();
 kirim_sensorpir();
 kirim_data();
 kirim_datagas()

```

```

}

```

**//koding buzzer**

```

delay(1000); // 10fps

```

```

} catch (exception e) {
    cout<<"error"<<endl;
}

```

```

}
return (0);

```

```

}

```

# PERANCANGAN SISTEM PENGENDALI PERANGKAT ELEKTRONIK OTOMATIS BERBASIS IOT

Ciksadan <sup>1)</sup>, Suroso <sup>2)</sup>, Yuris Ramadhona <sup>3)</sup>

<sup>1),2),3)</sup> Jurusan Teknik Elektro Program Studi Teknik Telekomunikasi, Politeknik Negeri Sriwijaya Jl. Sriwijaya Negara, Bukit Besar Palembang  
E-mail: [cik\\_sadan@yahoo.com](mailto:cik_sadan@yahoo.com) <sup>1)</sup>

## ABSTRACT

*Controlling electronic devices manually often makes the user forget even sometimes lazy to turn off the device when not in use anymore. It can result in waste of electrical energy and even short circuit. In addition, the rampant cases of empty home theft often make homeowners anxious when traveling far away. With technological advances, manual operation of electronic devices can be transformed into an IoT-based automated control system aimed at making controlling more effective and efficient by using the PIR sensor for human detection and DHT11 sensor for temperature detection as the input. Will be managed by the Raspberry Pi and the output is an auto-controllable electronic device and an e-mail notification to the homeowner.*

*Keywords : Raspberry Pi, PIR Sensor, DHT11 Sensor, IoT, E-mail*

## ABSTRAK

Pengendalian perangkat elektronik secara manual sering membuat pengguna lupa bahkan kadang malas untuk mematikan perangkat ketika tidak digunakan lagi. Hal tersebut dapat mengakibatkan pemborosan energi listrik bahkan konsleting listrik. Selain itu, maraknya kasus pencurian rumah kosong kerap membuat cemas pemilik rumah ketika bepergian jauh. Dengan kemajuan teknologi, pengoperasian perangkat elektronik secara manual dapat diubah menjadi sistem kontrol otomatis berbasis IoT yang bertujuan menjadikan pengontrolan lebih efektif dan efisien dengan menggunakan sensor PIR untuk mendeteksi manusia dan sensor DHT11 untuk deteksi suhu sebagai input masukan yang akan dikelola oleh *Raspberry Pi* dan menghasilkan output perangkat elektronik yang dapat dikendalikan otomatis dan notifikasi berupa *e-mail* kepada pemilik rumah.

Kata kunci : Raspberry Pi, Sensor PIR, Sensor DHT11, IoT, E-mail

## PENDAHULUAN

Pengendalian perangkat elektronik secara manual sering membuat pengguna lupa bahkan kadang malas untuk mematikan perangkat ketika tidak digunakan lagi[1]. Selain menjadi penyebab pemborosan energi listrik dan juga tagihan listrik, perangkat elektronik yang terus menyala juga sering mengakibatkan konsleting listrik.

Masalah yang sering muncul karena pengendalian perangkat secara otomatis bukan hanya pemborosan energi listrik dan juga konsleting listrik tetapi juga banyaknya kasus pencurian yang terjadi ketika sebuah rumah dalam keadaan kosong. Biasanya, kejadian itu terjadi ketika pemilik rumah lupa bahkan tidak menyalakan lampu ketika mereka pergi di malam hari dan lampu tetap menyala di siang hari. Inilah yang menyebabkan si pencuri mengetahui bahwa pemilik rumah tidak ada di sana, sehingga mereka memiliki kesempatan untuk mencuri di rumah[2].

Dengan kemajuan teknologi, operasi manual dapat diubah menjadi sistem kontrol otomatis berbasis IoT yang bertujuan untuk membuat kontrol berjalan lebih efektif dan efisien. Sehingga dapat menghindari lampu dan kipas yang berjalan sia-sia jika tidak ada aktivitas di dalamnya untuk menghemat lebih banyak listrik dan tagihan listrik yang dikeluarkan dan juga pemilik rumah dapat mengontrol jarak jauh perangkat elektronik ketika mereka tidak di rumah untuk mengurangi kasus pencurian ketika rumah itu kosong.

Pemanfaatan teknologi pengendalian perangkat elektronik otomatis telah berhasil dilakukan pada beberapa penelitian sebelumnya dengan penggunaan mikrokontroler dan proses monitoring yang berbeda. Pada

penelitian Galoeh Otomo dan Wildian (2013) dengan judul "Sistem Kontrol Penyalan Lampu Ruang Berdasarkan Ada Tidaknya Orang di Dalam Ruang", sistem tersebut bekerja berdasarkan program yang diolah menggunakan mikrokontroler AT8951S[3]. Lalu pada penelitian Albert Gifson dan Slamet (2009) dengan judul "Sistem Pemantau Ruang Jarak Jauh Dengan Sensor *Passive Infrared* Berbasis Mikrokontroler At89s52" sistem tersebut menggunakan mikrokontroler AT89S52 dalam mengolah datanya[4].

Pada penelitian ini, sensor yang digunakan untuk pengoperasian perangkat secara otomatis adalah sensor PIR, sedangkan untuk mendeteksi suhu ruangan digunakan sensor DHT11. Sensor pir adalah sensor panas jenis *pyroelectric* yang dapat merespon perubahan suhu panas tubuh manusia menggunakan inframerah[5] sedangkan Sensor DHT11 merupakan sensor dengan kalibrasi sinyal digital yang mampu memberikan informasi suhu dan kelembaban[6]. Suhu tubuh manusia dan suhu ruangan akan masuk sebagai input respon yang akan dikelola oleh *raspberry pi* sehingga menghasilkan output berupa perangkat elektronik yang menyala. Data yang telah diproses pada raspberry akan masuk ke database yang dibuat menggunakan bahasa pemrograman yang kemudian juga akan diteruskan untuk dikirim ke e-mail pengguna.

## TINJAUAN PUSTAKA

### ***Internet Of Things (IoT)***

*Internet Of Things (IoT)* adalah kemampuan dalam mengirimkan data tanpa memerlukan koneksi antara manusia ke komputer atau manusia ke manusia[7]. Dalam Penelitian ini, IoT berfungsi menghubungkan perangkat ke web server. Sehingga mempermudah

pemilik rumah untuk mengawasi maupun mengendalikan perangkat elektronik dari jarak jauh.

## Raspberry Pi

*Raspberry Pi* merupakan komputer dalam satu singleboard yang dikembangkan oleh yayasan *Raspberry Pi* di Inggris. *Raspberry Pi* memiliki dua model yaitu model A dan model B dengan seri 1, 2 dan 3[8]. Seri yang digunakan dalam penelitian ini adalah *Raspberry Pi 3* model B+ yang merupakan seri terbaru. *Raspberry Pi 3* model B+ memiliki prosesor 64-bit yang kini memiliki clockspeed maksimum 1.4 GHz (sebelumnya 1.2 GHz pada Pi 3B), memiliki Gigabit Ethernet (support PoE) yang tentu jauh lebih kencang dari versi sebelumnya, memiliki heatsink pada prosesor untuk distribusi panas yang lebih baik, serta mendukung dual band WLAN 5 GHz dan 2.4 GHz[9]. Bagian utama dari *Raspberry Pi* adalah *processor*. Setiap *Raspberry Pi* memiliki *BCM2835 Chip Broadcom*. Chip ini memiliki *clock speed* 700MHz dan merupakan sistem 32-bit [10]. Dalam koneksi ke jaringan bisa menggunakan port Ethernet/LAN atau pada *Raspberry Pi 3* model B sudah dilengkapi dengan *wifi built-in* [11].



Gambar 1. Raspberry Pi 3 B+  
(Sumber : Krisniawan, Adi, 2015)

## Sensor PIR

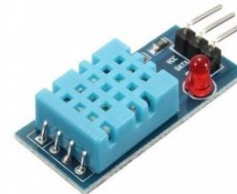
Sensor PIR merupakan sensor berbasis inframerah. Sensor PIR hanya merespon dari pancaran inframerah setiap benda yang terdeteksi olehnya[12]. Dengan adanya sensor PIR, maka akan terdeteksi keberadaan manusia yang bergerak melewatinya[13].



Gambar 2. Sensor PIR  
(Sumber : Riskiono, S.D, 2018)

## Sensor DHT11

Sensor yang digunakan untuk mengukur suhu udara adalah sensor DHT11. Sensor ini tergolong smart sensor karena memiliki ADC yang telah terintegrasi dalam sensor tersebut. Sensor DHT11 merupakan sensor yang banyak digunakan karena kemudahannya dan kelebihanannya dapat mengukur nilai suhu dan kelembaban udara sekaligus[14]. DHT11 memiliki fitur kalibrasi yang sangat akurat. Koefisien kalibrasi ini disimpan dalam OTP program memory, sehingga ketika internal sensor mendeteksi sesuatu suhu atau kelembaban, maka module ini membaca koefisien sensor tersebut. Ukurannya yang kecil, dengan transmisi sinyal hingga 20 meter, membuat produk ini cocok digunakan untuk banyak aplikasi-aplikasi pengukuran suhu dan kelembaban[6].



Gambar 3. Sensor DHT11  
(Sumber : Eka, A.M.Y., 2013)

## Web Server

Aplikasi web adalah suatu jenis aplikasi komputer yang dapat diakses dengan web browser. Web Server adalah server yang melayani permintaan aplikasi web[15]. Web server ditulis dengan bahasa pemrograman PHP dan database MySQL.

## Bahasa Pemrograman

Merupakan teknik instruksi standar untuk memberikan perintah kepada komputer yang memungkinkan programmer dapat menentukan secara persis data mana yang akan diolah dan

dijalankan oleh komputer, bagaimana penyimpanannya, dan menentukan langkah apa yang akan diambil dari berbagai jenis situasi yang telah ditentukan[16]. Terdapat beberapa bahasa pemrograman yang sering digunakan, antara lain : *Hypertext Preprocessor*(PHP), *Hypertext Markup Language*(HTML), Bahasa C++, Javascript dan Css, serta MYSQL.

### Database

*Database* adalah sebuah tempat penyimpanan yang besar dimana terdapat kumpulan data yang tidak hanya berisi data operasional tetapi juga deskripsi data[17].

### Sublime Text

*Sublime Text* merupakan salah satu *text editor* yang sangat *powerful* yang dapat meningkatkan produktivitas dan mengembangkan kualitas kode yang tinggi[17].

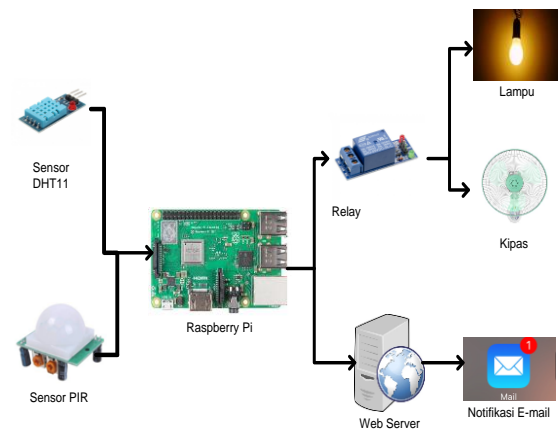
### E-mail

*Email* adalah surat elektronik yang dikirim dengan menggunakan internet. Penggunaan *electronic mail* (surat elektronik) sebagai media komunikasi yang ditunjang oleh situs internet seperti Yahoo, Google, Google Mail, MSN, Wordpress, dan situs lainnya. Banyaknya situs internet dalam jaringan internasional network menunjukkan bahwa orang mau melakukan komunikasi karena dengan komunikasi orang dapat beraktivitas dan meningkatkan kariernya[18].

## HASIL DAN PEMBAHASAN

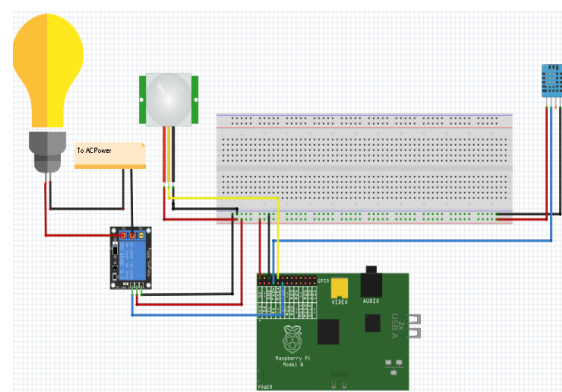
### Desaian Hardware

Langkah awal dalam membuat perangkat keras sebuah sistem adalah melakukan pemilihan komponen dan membuat blok diagram. Hal ini dilakukan agar komponen yang dipilih memiliki spesifikasi sesuai dengan yang dibutuhkan. Berikut merupakan blok diagram sistem :



Gambar 4. Blok Diagram Sistem

Pada gambar 4 merupakan blok diagram sistem pengendali dan monitoring yang akan dibangun. Sistem ini menggunakan Raspberry Pi 3 B+ sebagai otak utama dalam mengendalikan sistem ini. Sensor PIR digunakan sebagai input yang mendeteksi keberadaan manusia disekitarnya sedangkan sensor dht11 digunakan untuk mendeteksi suhu disekitar ruangan. Radiasi panas tubuh manusia dan suhu ruangan akan diterima sensor PIR dan sensor DHT11 sebagai input masukan yang akan dikelola oleh *Raspberry Pi* sehingga akan menghasilkan 2 output yaitu relay untuk menghidup dan mematikan perangkat elektronik, serta *web server* yang berguna untuk menyimpan data di web dan mengirimkan notifikasi berupa *e-mail* kepada pemilik rumah. Berikut merupakan rangkaian keseluruhan :

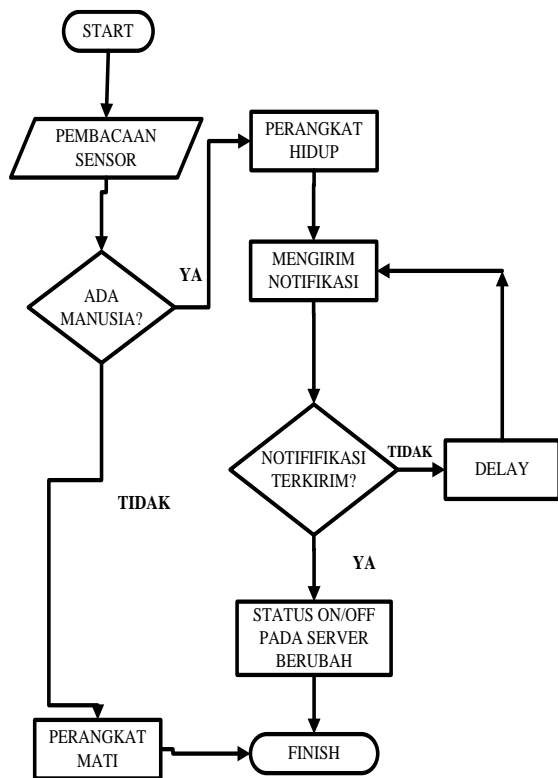


Gambar 5. Rangkaian Keseluruhan

Pada gambar 5 menunjukkan pemasangan komponen yang harus terhubung satu sama lain dengan port-port yang ditentukan berdasarkan datasheet berbagai komponen.

## Desain Software

Desain software diperlukan untuk menghubungkan sistem ke internet agar dapat menyimpan dan mengirimkan data melalui e-mail. Pada pembuatan desain software, terdapat koding-koding yang dibuat menggunakan bahasa c++ yang ditulis di netbeans yang akan diupload pada putty sehingga terprogram ke raspberry pi dan menjalankan program sesuai dengan diagram alir sistem. Berikut merupakan diagram alir sistem :



Gambar 6. Flowchart Sistem

Pada gambar 6 merupakan diagram alir sistem yang dimulai dengan pembacaan sensor. Jika terdeteksi keberadaan manusia, maka sensor akan mengirimkan input ke raspberry pi yang kemudian akan di proses oleh raspberry Pi sehingga akan menghasilkan output relay untuk menghidupkan perangkat. Jika perangkat sudah hidup, maka sistem akan mengirimkan notifikasi berupa e-mail kepada pemilik rumah. Jika notifikasi terkirim, maka status on/off pada web server akan berubah dan tugas sistem selesai. Sebaliknya, jika tidak terdeteksi manusia, maka

sistem tidak akan menyalakan perangkat elektronik.

Selain membuat diagram alir, *desain software* juga diperlukan untuk membuat database yang digunakan untuk menyimpan data yang keluar dari perangkat keras, yang digunakan sebagai acuan apakah sistem berjalan dengan baik atau tidak. Pembuatan database dilakukan dengan membuat program pada sublime text menggunakan bahasa c++ yang kemudian diupload pada web server.

## Pengujian Sistem

Hasil Keluaran pada penelitian ini adalah dapat menghidupkan perangkat elektronik secara otomatis dan mengirimkan notifikasi melalui e-mail. Selain itu, pemilik rumah juga dapat mengatur on/off perangkat melalui jarak jauh pada *web server* asalkan terhubung dengan jaringan internet.

Tabel 1. Pengujian Sensor

Jarak Objek	Lama Respon Sensor	Respon
1 meter	4 detik	Nyala
2 meter	6 detik	Nyala
3 meter	10 detik	Nyala
4 meter	20 detik	Nyala
5 meter	60 detik	Nyala

Pada pengujian sensor, terdapat 5 jarak objek dimulai dari 1 meter hingga 5 meter. Lama respon sensor adalah delay/waktu yang diperlukan sensor untuk mendeteksi adanya objek yang melewatinya dalam jangkauan jarak



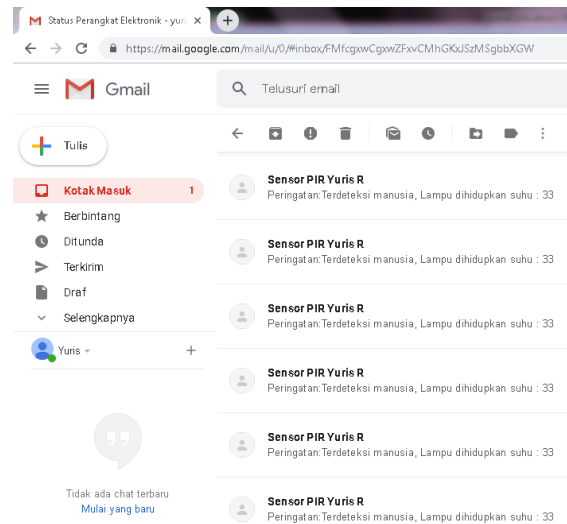
yang ditentukan sehingga akan menghasilkan output terhadap respon. Dari pengujian tersebut, dapat diketahui bahwa sensor pir memiliki sensitivitas tinggi terhadap objek yang melewatinya.

Tampilan data yang masuk saat pengujian data pada database adalah seperti berikut :

ID	Waktu	Status_PIR	Suhu
201	2019-05-31 15:33:45	Tidak Terdeteksi Manusia	33
202	2019-05-31 15:33:47	Tidak Terdeteksi Manusia	33
203	2019-05-31 15:33:48	Tidak Terdeteksi Manusia	33
204	2019-05-31 15:33:49	Tidak Terdeteksi Manusia	33
205	2019-05-31 15:33:51	Terdeteksi Manusia	33
206	2019-05-31 15:33:52	Terdeteksi Manusia	33
207	2019-05-31 15:33:53	Tidak Terdeteksi Manusia	33
208	2019-05-31 15:33:55	Tidak Terdeteksi Manusia	33
209	2019-05-31 15:33:56	Tidak Terdeteksi Manusia	33
210	2019-05-31 15:33:57	Tidak Terdeteksi Manusia	33
211	2019-05-31 15:33:58	Tidak Terdeteksi Manusia	33
212	2019-05-31 15:33:59	Terdeteksi Manusia	33
213	2019-05-31 15:34:01	Terdeteksi Manusia	33
214	2019-05-31 15:34:04	Tidak Terdeteksi Manusia	33
215	2019-05-31 15:34:05	Tidak Terdeteksi Manusia	33
216	2019-05-31 15:34:06	Tidak Terdeteksi Manusia	33
217	2019-05-31 15:34:07	Tidak Terdeteksi Manusia	33
218	2019-05-31 15:34:09	Tidak Terdeteksi Manusia	33
219	2019-05-31 15:34:10	Tidak Terdeteksi Manusia	33
220	2019-05-31 15:34:11	Terdeteksi Manusia	33
221	2019-05-31 15:34:12	Terdeteksi Manusia	33

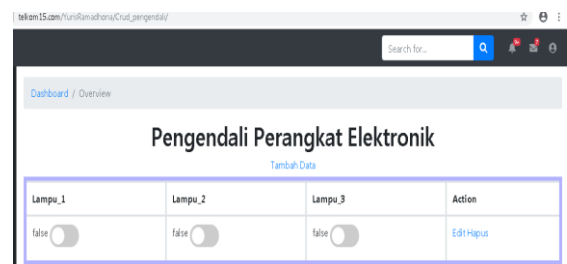
Gambar 7. Data masuk pada web server

Dari data diatas, terdapat status pir ketika mendeteksi keberadaan manusia dan juga status sensor dht11 yang mendeteksi suhu ruangan. Sedangkan untuk tampilan notifikasi e-mail dengan isi pesan berupa peringatan adanya manusia adalah sebagai berikut :



Gambar 8. E-mail masuk saat terdeteksi manusia

Adapun perubahan status on/off pada web server ketika lampu hidup dan menyala adalah :



Gambar 9. Tampilan Status Lampu off pada web server



Gambar 10. Tampilan Status On pada web server

## KESIMPULAN DAN SARAN

### Kesimpulan

Berdasarkan hasil dan analisis yang diperoleh maka dapat disimpulkan bahwa:

1. Sistem pengendali perangkat elektronik ini berjalan dengan baik. Sensor PIR dapat mendeteksi objek dengan jarak maksimal 5 meter, tetapi

terkadang pada jarak tersebut juga sering tidak terdeteksi. Sehingga jarak deteksi maksimal yang efektif adalah 3 meter.

2. Sensor DHT11 berfungsi dengan baik dalam mendeteksi suhu ruangan.
3. Setelah sensor mendeteksi keberadaan manusia, maka sistem akan mnghidupkan lampu dan mengirimkan notifikasi ke email pemilik rumah.
4. Penggunaan Raspberry pi mempermudah pengguna dalam mengintegrasikan perangkat dengan software layaknya fungsi komputer dengan jumlah ram dan penyimpanan yang lebih tinggi dibanding arduino.
5. Dengan adanya sistem ini, dapat mempermudah manusia dalam pengendalian perangkat elektronik dari jarak jauh dengan bantuan internet sehingga mengurangi pemborosan energi listrik dan resiko pencurian rumah kosong.

### **Saran**

Dari penelitian yang telah dilakukan, sebaiknya penelitian selanjutnya untuk menggunakan sensor yang sensitivitas terhadap manusia yang lebih tinggi lagi dibanding makhluk hidup lainnya. Karena selain dapat mendeteksi manusia, sensor pir juga memiliki sensitivitas tinggi terhadap gerakan hewan yang lewat didepannya.

## REFERENSI

1. Prihatmoko, Dias. 2015. "Perancangan Sistem Kontrol Lampu Pada Gedung Fakultas Sains Dan Teknologi Unisnu Jepara Menggunakan Wireless Sensor Network" dalam Jurnal Disprotek. Volume 6 (hlm. 49-50).
2. Mawardi, Isal. 2019. "Rumah Kosong di Bekasi di Bobol Maling" dalam [https://news.detik.com/berita/d-4540850/rumah-kosong-di-bekasi-dibobol-maling-1-unit-mobil-dibawa-kabur?\\_ga=2.47278178.454993099.1558621317-2055737753.1558621317](https://news.detik.com/berita/d-4540850/rumah-kosong-di-bekasi-dibobol-maling-1-unit-mobil-dibawa-kabur?_ga=2.47278178.454993099.1558621317-2055737753.1558621317) (diakses, 23 mei 2019).
3. Otomo, Galoeh dan Wildian .2017. "Sistem Kontrol Penyalaan Lampu Ruang Berdasarkan Pendeteksian Ada Tidaknya Orang Di Dalam Ruangan" dalam jurnal FISIKA UNAND. Volume 2 Nomor 4 (hlm. 255-258).
4. Gifson, Albert Dan Slamet. 2009. "Sistem Pemantau Ruang Jarak Jauh Dengan Sensor *Passive Infrared* Berbasis Mikrokontroler At89s52" Dalam Jurnal Telkomnika. Volume 7 Nomor 3 (Hlm. 201-205).
5. Ahadiah, Siti., Muharnis, Agustawan. 2017. "Implementasi Sensor Pir Pada Peralatan Elektronik Berbasis Microcontroller" dalam JURNAL INOVTEK POLBENG. Volume 7 (hlm. 29-30).
6. Eka, A.M.Y., dan Hari Wibawanto. 2013. Sistem Pengamatan Suhu dan Kelembaban Pada Rumah Berbasis Mikrokontroler ATmega8 dalam JURNAL TEKNIK ELEKTRO UNNES. Volume 5 Nomor 1 (hlm.15-16).
7. Ramayani, Tiara dkk. 2018. "Penerapan IoT (Internet Of Things) Untuk Pencegahan Dini Terhadap Kejahatan Begal" dalam JURNAL RESTI. Volume 3 Nomor 2 (hlm. 628).
8. Krisniawan, Adi. 2015. "Perancangan Sistem Keamanan Ruangan Menggunakan Raspberry Pi" dalam e-proceeding of Engginering. Volume 2 Nomor 2 (hlm. 3823).
9. Christian, Fendi. 2017. "Modul Pembelajaran Raspberry Pi" dalam Tugas Akhir Teknik Elektro, Universitas Sanata Dharma (hlm. 5-6).
10. Shadiq, H.M, Sudjadi, Darjat. 2014. "Perancangan Kamera Pemantau Nirkabel Menggunakan Raspberry Pi Model B. TRANSIENT Volume 3 no 4 (hlm. 546-551).
11. Susanti, E., & Triyono J. 2016. "Prototype Alat Iot (Internet Of Things) Untuk Pengendali Dan Pemantau Kendaraan Secara Realtime" dalam Simposium Nasional RAPI XV (hlm. 401-407).
12. Sekar, A.K.R., Subali. "Sistem Keamanan Rumah Otomatis Menggunakan Sensor Pir, Sensor Suhu, Sensor Gas Yang Terhubung Dengan Telepon Seluler Berbasis Mikrokontroler Atmega8 Dan Mikrokontroler Atmega162 Dengan Backup Daya" dalam JURNAL GEMA TEKNOLOGI. Volume 17 Nomor 2 (hlm. 86-87).
13. Riskiono, S.D dkk. 2018. "Implementasi Sensor Pir Sebagai Alat Peringatan Pengendara Terhadap Penyeberang Jalan Raya" dalam JURNAL MIKROTIK. Volume 8 Nomor 1 (hlm. 57-58).

14. Sari, I.A, Anik N.H, Dyah L. 2018. "Smart Greenhouse Sebagai Media Pembibitan Kentang Granola Kembang Berbasis Mikrokontroler" dalam Proc. Seminar Nasional Teknologi Elektro Terapan Volume 2 no 1 (hlm. 105-110).
15. Kadir, Abdul. 2009. *From Zero to A Pro : Membuat Aplikasi Web dengan PHP dan Database MySQL*. Yogyakarta : Andi Offset.
16. Hidayat, Harri., dkk. 2015. "Pengembangan Learning Management System untuk Bahasa Pemrograman PHP" dalam Jurnal Ilmiah Core IT (hlm. 21).
17. Pahlevi, Omar., dkk. 2018. "Sistem Informasi Inventori Barang Menggunakan Metode *Object Oriented* Di Pt. Livaza Teknologi Indonesia Jakarta" dalam JURNAL PROSISKO. Volume 5 Nomor 1 (hlm. 28-30).
18. Anhar Mawarsih. (2014). *Pengaruh Electronic Mail Sebagai Media Komunikasi Terkadap Mengerjakan Tugas Kuliah Mahasiswa* dalam E-journal Ilmu Komunikasi. Volume 2 Nomor 1 (hlm. 337-338).