

## Otomasi sistem kelistrikan menggunakan algoritma a-star berbasis internet of things

Januar Al-Amin<sup>1</sup>, Evans Fuad<sup>2</sup>, Muhammad Waqi Azizi<sup>3</sup>

<sup>1,2,3</sup>Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau

Jalan KH. Ahmad Dahlan No. 88, Pekanbaru, Riau

e-mail : <sup>1</sup>januaralamin@umri.ac.id, <sup>2</sup>evansfuad@umri.ac.id, <sup>3</sup>mwaqiazizi@student.umri.ac.id

### Abstrak

Meluasnya penerapan mikrokontroler dalam kehidupan kita sehari-hari mendorong penerapan yang lebih dalam, seperti mesin cuci piring pintar, kendali kendaraan pintar, serta perangkat rumah pintar yang merupakan ide-ide menarik untuk diterapkan. Aktivitas perkuliahan yang berlangsung dari pagi hingga malam hari mengindikasikan intensitas penggunaan energi listrik yang tinggi. Hal ini dikarenakan pembagian jadwal perkuliahan yang ada, didapatkan rata-rata kejadian kondisi perangkat kelistrikan ditinggalkan dalam kondisi hidup setelah perkuliahan selesai adalah 6-10 ruangan per hari per lokasi antara kampus 1 dan kampus 2. Tingginya tingkat pemborosan energi oleh ruangan yang ditinggalkan dalam kondisi perangkat listrik yang tetap hidup setelah aktivitas perkuliahan selesai, mendorong penerapan penggunaan algoritme heuristik sebagai kerangka kerja sistem pengendalian perangkat listrik otomatis. Sistem ini akan membantu penghematan biaya penggunaan energi listrik. Didukung oleh algoritme heuristik A-star, penerapan mikrokontroler Raspberry Pi 3 sebagai sistem kontrol utama sistem otomasi kelistrikan rumah/bangunan memberikan solusi penghematan konsumsi dan efektivitas penggunaan listrik.

**Kata kunci:** mikrokontroler, algoritme heuristik, internet of things, efektivitas dan penghematan listrik

### Abstract

The widespread application of microcontrollers in our daily lives is a deeper application, such as smart dishwashers, smart vehicle access, and smart home devices which are interesting ideas to implement. Lecture activities that take place from morning to night indicate the high intensity of electrical energy usage. This is due to the distribution of the existing lecture schedule. It is found that the average incidence of electrical equipment left in living conditions after the lecture is complete is 6-10 rooms per day per location between campus 1 and campus 2. The high level of waste of energy by the room left in the condition electrical devices that remain alive after lecture activities are completed, encouraging the application of the use of heuristic algorithms as a framework for automated electrical appliance control systems. This system will help save the cost of using electricity. Supported by the A-star heuristic algorithm, the application of the Raspberry Pi 3 microcontroller as the main control system of home / building electrical automation systems provides solutions of saving consumption and the effectiveness of electricity usage.

**Keywords:** microcontroller, heuristic algorithm, effectiveness and saving electricity

### 1. Pendahuluan

Meluasnya penerapan mikrokontroler dalam kehidupan kita sehari-hari mendorong penerapan yang lebih dalam, seperti mesin cuci piring pintar, kendali kendaraan pintar, serta perangkat rumah pintar yang merupakan ide-ide menarik untuk diterapkan. Permasalahan pada ruang kelas kampus 1 dan 2 UMRI adalah kondisi perangkat listrik terutama lampu yang ditinggalkan dalam kondisi hidup setelah tidak ada lagi kegiatan di ruangan.

---

Dari pengamatan di lapangan didapatkan rata-rata jumlah ruangan yang ditinggalkan dalam kondisi perangkat listrik hidup adalah 6-10 ruangan per hari. Hal ini menunjukkan penggunaan perangkat listrik di ruangan yang belum dikelola dengan baik dan indikasi pemborosan energi listrik.

Wibisono dan Bayhaki [1] membahas pendekatan otomasi lampu jalan menggunakan *ad-hoc network* berbasis *wireless connectivity*; Leech et al. [2] menekankan pendekatan optimisasi *memory* sistem pemantauan kondisi ruangan yang bekerja menggunakan algoritma Bayesian; Cynthia et al. [3] membahas sistem otomasi perangkat listrik kelas menggunakan mikrokontroler Arduino tanpa adanya algoritma tertentu, menekankan pada ketiadaan aktivitas selama 10 menit didalam kelas sebagai sebagai penentu keputusan; dan Lee et al. [4] membahas sistem *tracking* kendaraan berbasis mikrokontroler yang dipasang pada mesin kendaraan dengan mengirimkan data posisi perangkat GPS/GSM/GPRS yang terhubung dengan mikrokontroler ke *database* sistem monitoring.

Dengan menggunakan sejumlah penelitian di atas sebagai bahan rujukan, dan inti permasalahan yang dibahas di awal, penelitian ini mengajukan sebuah sistem otomasi kontrol listrik yang menggunakan mikrokontroler sebagai perangkat utama sistem berbasis sensor PIR. Penggunaan mikrokontroler Raspberry Pi 3 Model B mengontrol beberapa perangkat sekaligus ; sensor PIR, *database*, dan perangkat elektronik (lampu, stop kontak) dengan mikrokontroler sebagai server sistem otomasi. Penelitian ini berfokus pada penghitungan *timer* dari data pembacaan sensor dalam kurun waktu tertentu untuk mengurangi beban pemrosesan algoritme. Sehingga pemrosesan skala kecil tidak menyebabkan *overload* kapasitas sumber daya mikrokontroler[5] dan proses penghitungan lebih cepat dilakukan.

Sistem otomasi bekerja menggunakan *relay* yang berfungsi sebagai saklar mekanik arus listrik AC dan algoritme heuristik A-star sebagai *tools* untuk menghitung data dari hasil pembacaan sensor. Hasil perhitungan dari algoritme akan menentukan *delay time* yang dikirimkan ke *relay* untuk mematikan perangkat listrik secara otomatis (sistem lampu) dengan menghitung data penggunaan lampu yang tersimpan didalam *database*.

## 2. Metode Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yaitu studi literatur, analisa, dan perancangan. metode penelitian dapat dilihat pada bagan dibawah ini :

### 2.1 Studi Literatur

Pada tahap ini peneliti mencari sumber informasi terkait konsep yang dipakai pada sistem *Internet of Things*, konsep penggunaan sensor PIR, konsep *relay* sebagai saklar listrik, konsep algoritme A-star, dan penggunaan database pada server berbasis mikrokontroler. Literatur yang digunakan diperoleh dari jurnal-jurnal, buku, serta artikel penelitian terdahulu terkait bidang mikrokontroler.

#### 2.1.1 Passive Infrared Sensor (PIR Sensor)

Sensor pasif inframerah (PIR) adalah sensor untuk mengukur gelombang inframerah yang memancar dari objek bergerak. Sensor PIR mendeteksi sumber panas dan objek bergerak seperti tubuh manusia dan umumnya bekerja pada temperatur 35 – 37 derajat Celsius untuk pengaturan sensitifitas panas [6].

Sensor PIR yang digunakan pada penelitian ini adalah HC SR-505 dan HC SR-501 dengan luas jangkauan area +/- 5 meter (bisa diatur menggunakan switch sensitivitas) dengan bidang area 110 derajat didepan sensor.

#### 2.1.2 Raspberry Pi 3

Raspberry Pi adalah perangkat mikrokontroler yang berfungsi sebagai unit kontrol untuk sejumlah sensor, *motor*, *relay*, atau perangkat elektronik lainnya yang terhubung melalui kabel ataupun nirkabel dengan papan pin Raspberry Pi [7]. Raspberry Pi menggunakan masukan (*input*) data dari sensor untuk menghasilkan keluaran (*output*) perintah yang sesuai. Raspberry

Pi 3 Model B didukung *wireless network* yang memungkinkan pengendalian via *remote* dan penggunaan sebagai sistem server mikrokontroler [8].

### 2.1.3 Relay

*Relay* adalah saklar listrik non-konvensional. *Relay* menggunakan kumparan elektromagnetik untuk beroperasi sebagai saklar mekanik, beberapa varian *relay* menggunakan *solid-state* sebagai prinsip operasinya [9].

*Relay* banyak digunakan sebagai alat pengendali sirkuit dengan sinyal berdaya rendah. Dalam telekomunikasi, *relay* digunakan sebagai penguat daya untuk menreproduksi dan mengirimkan ulang sinyal dari satu sirkuit ke sirkuit lain. Dalam teknologi elektronik, *relay* secara khusus digunakan sebagai operator perhitungan logika.

### 2.1.4 Algoritme A-star

Algoritme A-star adalah algoritme pencarian heuristik berdasarkan fungsi biaya yang disajikan pada rumus dibawah ini,

$$f(x) = g(x) + h(x),$$

dimana  $g(x)$  adalah jarak awal yang dibutuhkan untuk mencapai tempat akhir (F) dari tempat awal (S), dan  $h(x)$  (dikenal juga sebagai nilai heuristik) adalah jarak estimasi  $S(i)$  (jalur-jalur penghubung) menuju titik tujuan (F) [10]. Hasil dari perhitungan algoritme A-star memberikan jarak paling optimal yang dibutuhkan dari titik atau tempat awal (S) menuju titik atau tempat tujuan (F). Dalam penelitian ini, algoritme A-star akan digunakan untuk menghitung waktu optimal untuk mematikan dan menghidupkan perangkat lampu yang terhubung dengan server.

Algoritma ini dipakain untuk tujuan mengaplikasikan sebuah sistem kontrol listrik yang bekerja secara otomatis menggunakan mikrokontroler dan sensor dan mencegah penggunaan lampu di ruangan ketika tidak ada lagi kegiatan di ruangan tersebut

## 2.2 Analisa

Analisa dilakukan untuk menjabarkan identifikasi kerja, data pendukung, serta perancangan sistem yang akan dibuat. Penjelasan analisa yang dilakukan sebagai berikut :

1. Analisa fungsi dan bahasa yang akan digunakan pada pembuatan sistem yaitu fungsi pemanggilan *database*, fungsi pembacaan data sensor, dan fungsi eksekusi *switching relay* serta bahasa yang digunakan adalah Python 3
2. Analisa data yang dibutuhkan yaitu data pembacaan sensor, data kondisi perangkat lampu, data kondisi sensor, data waktu mati/hidup lampu, dan data *delay switching*
3. Analisa perancangan meliputi perancangan diagram blok, perancangan koneksi server dan sensor, serta perancangan sistem server yang bekerja pada mikrokontroler

## 2.3 Perancangan

Setelah proses analisa selesai, maka tahap selanjutnya yaitu perancangan sistem otomasi kelistrikan. Pada tahap perancangan meliputi diagram blok, arsitektur dan antarmuka sistem, interkoneksi sistem dan sensor, dan perancangan database yang digunakan pada sistem. Penelitian ini menggunakan MySQL sebagai *database* pada server.

---

### 3. Hasil dan Pembahasan

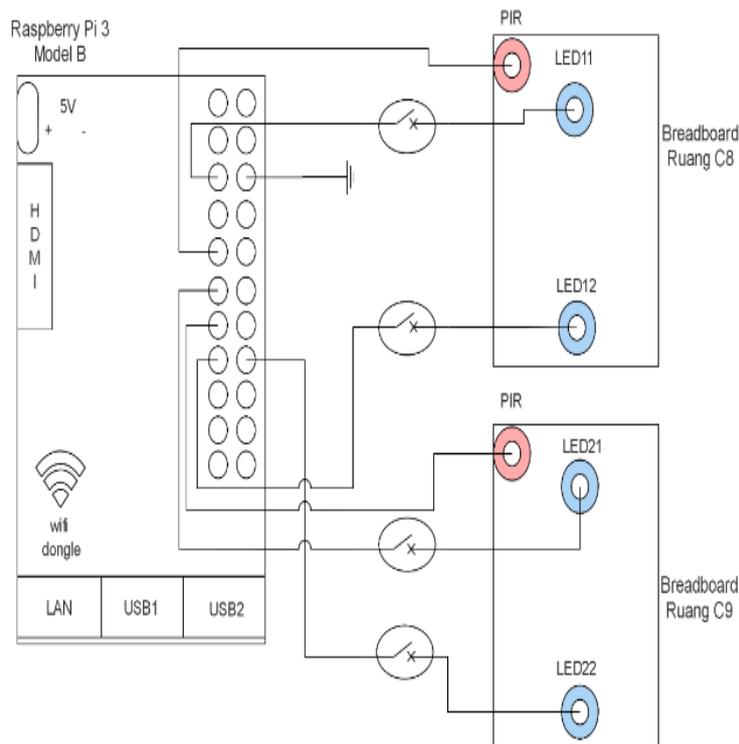
Analisis kebutuhan fungsional meliputi analisis kebutuhan perangkat lunak (*software*) dan perangkat keras (*hardware*) yang akan dibutuhkan yang disajikan melalui tabel berikut :

Tabel 1. Kebutuhan fungsional

Perangkat lunak ( <i>software</i> )	Perangkat keras ( <i>hardware</i> )
Redhat Fedora OS	PC Intel Pentium (R) 3.00 GHz x64 RAM DDR3 4 GB
Sublime text-editor	Raspberry Pi 3 + VGA-HDMI kabel display
Python language + Flask <i>micro-framework</i>	Sensor PIR, kabel Dupont male-female, switch, breadboard
Terminal	Wireless transceiver dongle

#### 3.1 Diagram Blok dan Perancangan Sistem

Skema rancangan pembuatan perangkat elektronik yang menjadi komponen utama di dalam sistem otomasi. Dalam pembuatan diagram skematik dimuat dengan detail jalur dan jenis komponen elektronik yang terdapat di dalam rangkaian utama perangkat elektronik. Berikut ini adalah diagram skematik alat yang akan dibuat



Gambar 1. Diagram Blok Perangkat

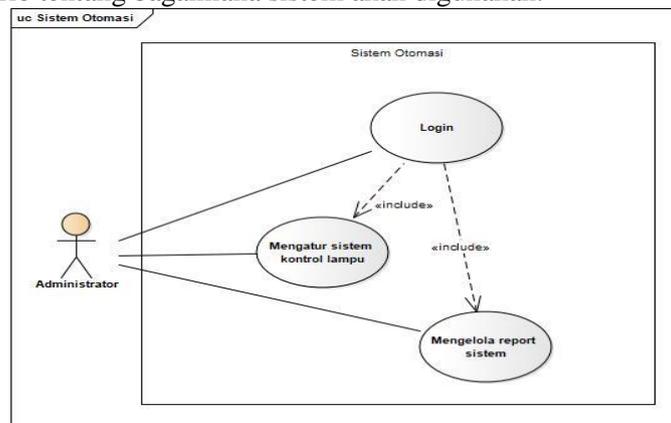
Disini dilakukan pemasangan kabel penghubung antar perangkat sistem; relay sebagai saklar AC/DC dengan *port input* terhubung ke pin 3,17, 22, 23 pada papan pin Raspberry Pi dan *port output* terhubung pada lampu diruang C8 dan C9; sensor PIR terhubung pada pin 18 dan 25; dan *ground* masing-masing perangkat dihubungkan dengan pin 6,9,14,20.

Relay yang digunakan disini adalah Tongling Relay 2 dengan batasan daya 10 W, cocok untuk perangkat lampu rumahan standar. Relay terhubung dengan papan Pi melalui breadboard yang dipasang 4 buah resistor 220 Ω ± 5% untuk masing-masing relay. Fungsi resistor disini untuk menghilangkan sisa daya pada masing-masing perangkat lampu setelah saklar dimatikan.

Perangkat lampu dipasang pada bagian kanan dan kiri ruangan, sehingga ada 2 buah perangkat lampu yang terhubung dengan sistem. Tiap lampu mempunyai spesifikasi daya 5W dan tegangan 220V. Perangkat elektronik lainnya dapat dihubungkan dengan papan Pi selama dalam batas daya 10 W dan tegangan 220V.

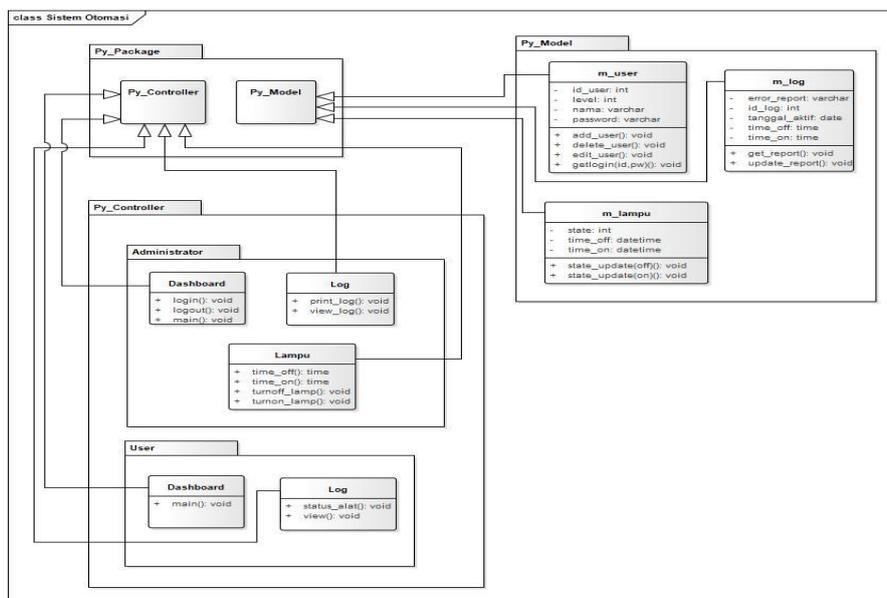
### 3.2 Pemodelan Sistem

Pemodelan sistem dilakukan menggunakan UML dimana terdapat *use-case diagram* yang menggambarkan interaksi antara pengguna sistem dengan sistem itu sendiri, dengan memberikan skenario tentang bagaimana sistem akan digunakan.



Gambar 2. Use Case Diagram Sistem

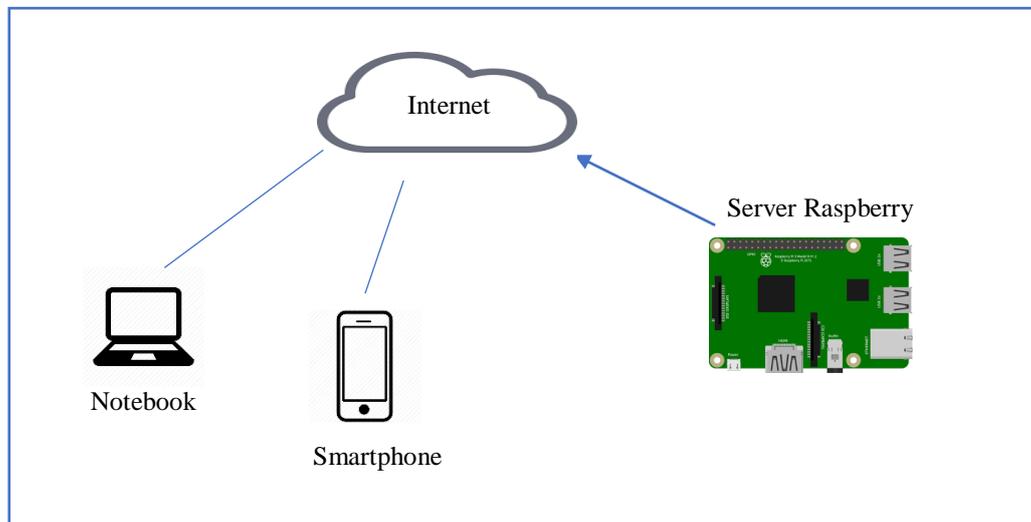
Dari *use-case* diatas dapat dilihat bahwa sistem terdiri atas 3 modul yaitu login, kontrol lampu, dan laporan aktif-tidaknya lampu. Dengan administrator sebagai *user* tunggal yang mengelola sistem beserta *datasenya*. Selain *use-case diagram*, pada pemodelan sistem ini juga menggunakan *class diagram* untuk memetakan interaksi antara modul kontrol sistem (*controller*) dengan modul database sistem (*model*) didalam bentuk *package* masing-masing modul.



Gambar 3. Class diagram sistem

### 3.3. Koneksi Alat, Internet, dan Gawai

Hubungan alat, internet sebagai media hubung, dengan gawai pengguna ditunjukkan pada gambar koneksi dibawah ini.



Gambar 4 Koneksi Alat, Internet, dan Gawai

Raspberry Pi sebagai perangkat utama sistem terhubung dengan sensor PIR dan *relay* saklar lampu melalui pin *board* yang terdapat pada papan rangkaian Raspberry Pi. Melalui internet, Raspberry Pi terhubung sebagai *server* monitoring yang dapat diakses melalui gawai yang memiliki akses internet seperti laptop dan *smartphone*. Untuk pengaturan Raspberry Pi sebagai *server*, digunakan *script* Python dibawah ini :

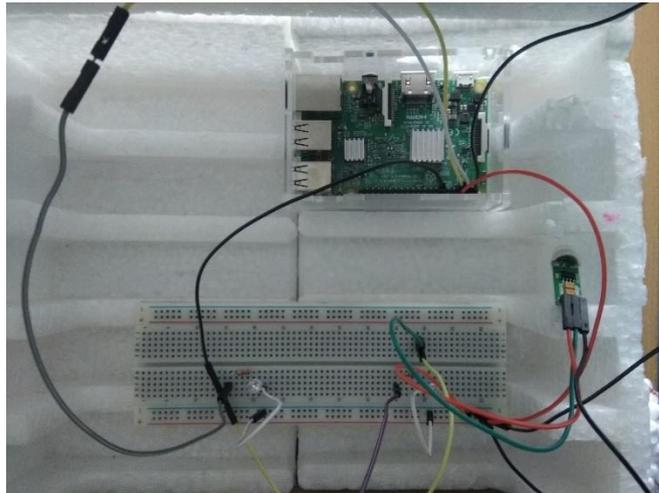
Tabel 2. *Script* Python Raspberry Pi sebagai *server*

```

from flask import Flask
app = Flask(__name__)
@app.route('/')
def index():
    # the main code here
if __name__ == '__main__':
    app.run(debug=True, port=100, host='192.168.43.136')
  
```

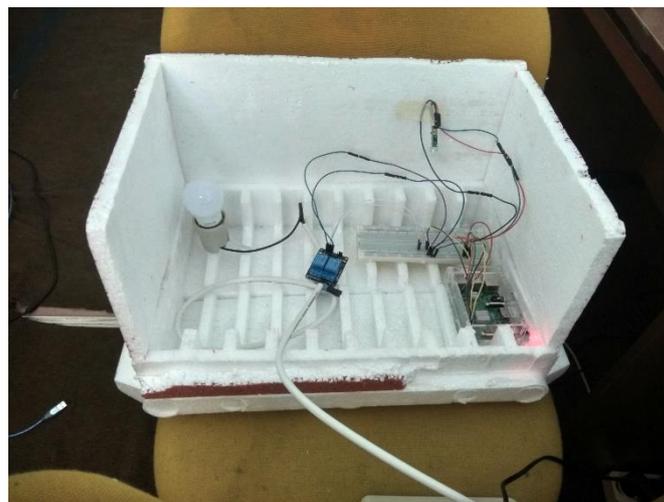
192.168.43.136 pada *port* 100. Sehingga ketika *server* akan aktif dan dapat diakses pada alamat 192.168.43.136:100 pada *browser*.

Berikut ini adalah rangkaian alat dimana Raspberry Pi sebagai *server* sistem telah terhubung dengan sensor PIR dan lampu melalui *breadboard*



Gambar 5. Rangkaian Alat

Ruang C8 dan C9 memiliki skema instalasi ruangan dimana terdapat 2 buah lampu di bagian depan dan belakang ruangan, 1 buah kipas angin, serta 2 buah colokan listrik di bagian depan dan samping kanan ruang. Maket alat dibuat menggunakan kardus dan *styrofoam* untuk mensimulasikan ruangan C8 dan C9. Pada tiap ruangan dipasang 2 buah lampu yang terhubung dengan *Raspberry Pi* melalui *breadboard*. Lampu di ruang C8 diberi label lampu 11 dan lampu 12, sementara lampu di ruang C9 diberi label lampu 21 dan lampu 22. Berikut ini adalah gambar maket alat yang dibuat :



Gambar 6. Maket Alat

### 3.4 Algoritma dan Cara Kerja Sistem

Sistem bekerja dengan uraian sebagai berikut :

- (i) Sensor deteksi keberadaan orang didalam ruang, tulis waktu mulai terbaca adanya deteksi orang sebagai variabel *time\_on* dan simpan ke database, tulis aksi deteksi sebagai 'SENSOR ON' pada sistem monitoring;
- (ii) Lanjutkan observasi sensor tiap interval 5 menit;
- (iii) Jika tidak ada lagi deteksi orang hasil pembacaan sensor PIR, tulis waktu tidak hadirnya tersebut sebagai variabel *time\_off* pada database dan sistem tampilkan kondisi tersebut sebagai 'SENSOR OFF' pada sistem monitoring;
- (iv) Sistem menghitung interval antara *time\_on* dan *time\_off* sebagai variabel *delta2*, atur waktu tunggu awal yaitu 10 detik (*wait\_default*), dengan hasil perhitungan *delta2* menghasilkan keputusan dibawah ini :

- a. Jika nilai delta 2 lebih dari 600 detik, maka waktu tunggu mematikan listrik dengan variabel wait akan diatur menjadi 600 detik;
  - b. Jika nilai delta2 lebih kecil dari 10 detik, maka variabel waktu tunggu wait diatur menjadi 10 detik;
  - c. Jika nilai delta2 berada pada *range*  $10 \leq \text{delta2} \leq 600$ , maka waktu tunggu wait diatur menjadi  $w = (\text{wait\_default}) + \text{delta2}$ ;
- (v) Sistem menyimpan aksi mematikan saklar sebagai timer\_off, simpan log waktunya, proses diulang terus hingga ada intervensi administrator atau sistem gagal membaca sensor.
- Otomasi diatas bekerja menggunakan script algoritme A-star yang dapat dilihat dibawah ini

:

Tabel 3. Perhitungan waktu tunggu menggunakan algoritme A-star

```

def selisih(wait):
    wait = 10
    delta2 = datetime.timedelta.total_seconds(time_off-time_on)
    # if delta2 > max waiting time
    if (delta2>600):
        wait = 600
    # if delta2 < min waiting time
    elif (delta2<10):
        wait = wait
    elif:
        wait = wait + delta2
    return wait

# system waiting time until switch lamp off
time.sleep(selisih)

```

yang bekerja sebagai berikut :

- (i) Atur  $g(x)$ (wait\_default) sebagai variabel awal waktu tunggu dengan nilai 10 detik;
- (ii) Simpan waktu masuk dan keluar orang, ambil data waktu tersebut dari database, didapatkan waktu keluar pada ruang C9 adalah 09:03:10 (time\_off) dan waktu masuk berikutnya 09:06:05 (time\_on);
- (iii) Hitung interval waktu tersebut sebagai  $h(x)$ (delta2), simpan dalam skala detik. Kita dapatkan hasilnya  $h(x) = 175$  secs;
- (iv) Atur  $f(x)$  sebagai waktu tunggu (wait var) untuk mematikan lampu. Menggunakan nilai delta 2 yang dihasilkan dari perhitungan sebelumnya (iii) dengan nilai wait maksimum 600 detik dan nilai minimum 10 detik ( $g(x)$ ), kita hitung nilai  $f(x)$  dengan maksimum dan minimum tersebut;
- (v) Hitung nilai  $f(x)$  dengan rumus dibawah ini:

$$f(x) = g(x) + h(x),$$

kita dapatkan hasil  $f(x)$  adalah 185 detik.

Karena hasil perhitungan masih dalam limit 10 – 600 detik, maka waktu tunggu diatur sesuai hasil perhitungan menjadi 185 detik.

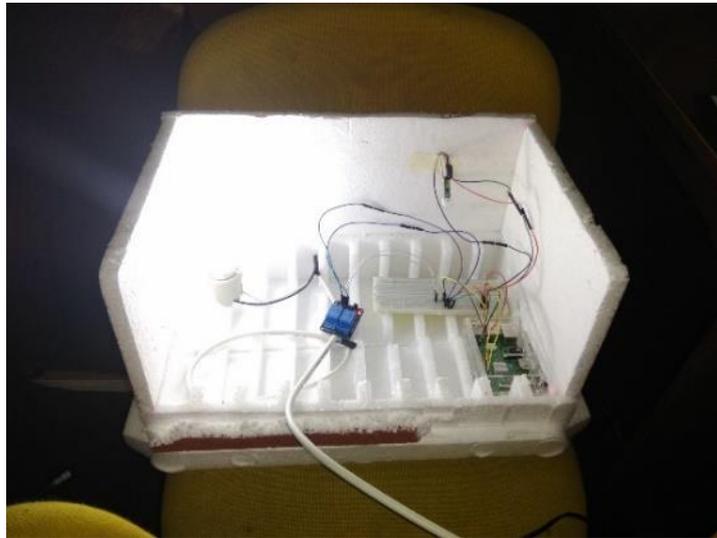
### 3.5 Hasil

Proses observasi dari data waktu yang dikumpulkan ke database sistem dilakukan dalam rentang waktu tertentu, kita mendapatkan hasil tunggu dan kalkulasi penghitungan berdasarkan algoritme ditampilkan pada tabel berikut ini,

Tabel 4. Perhitungan waktu tunggu menggunakan algoritme A-star

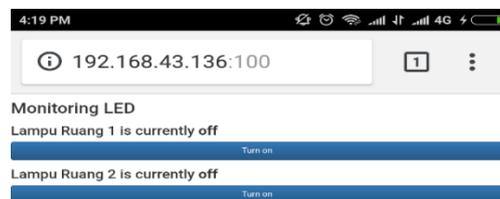
1	Report	Tanggal	
2	Lampu 1 dan 2 mati	2018-09-13 16:45:48	
3	Lampu 3 dan 4 hidup	2018-09-13 16:46:37	
4	Lampu 1 dan 2 hidup	2018-09-13 16:46:44	
5	Lampu 3 dan 4 mati	2018-09-13 16:47:04	
6	Lampu 1 dan 2 mati	2018-09-13 16:47:27	

Dari tabel diatas bisa ditarik kesimpulan bahwa tiap nilai interval lebih dari 600 detik akan membuat sistem untuk mengatur nilai timer\_off menjadi 600 detik dan tiap nilai interval kurang dari 10 detik membuat sistem mengatur nilai timer\_off menjadi 10 detik. Limitasi tersebut bekerja sebagai kontrol penyeimbang untuk tiap nilai interval diluar pemakaian normal.



Gambar 7. Demonstrasi sistem otomatis saat menhidupkan lampu pada maket

### 3.6 Monitoring Report



Gambar 8. Tampilan Monitoring Mobile

Sistem monitoring memberikan laporan mengenai tiap perubahan yang terjadi di dalam sistem, database dan koneksi antar perangkat, hingga kondisi perangkat sensor PIR sensors dan lampu di tiap ruangan. Jika terjadi *error*, seperti kegagalan pembacaan sensor PIR, sistem monitoring dapat digunakan untuk mengendalikan perangkat lampu secara *remote*.

#### 4. Kesimpulan

Tujuan penulis adalah membuat sistem otomasi yang bekerja menggunakan Raspberry Pi sebagai kontrol utama (server) yang terhubung dengan sensor PIR and perangkat lampu ruangan. Raspberry Pi yang merupakan mikrokontroler dapat digunakan sebagai perangkat kontrol otomatis untuk berbagai jenis perangkat elektronik rumahan selama dalam batasan daya dan tegangan papan elektronik Raspberry Pi. Penggunaan Raspberry Pi untuk sistem monitoring juga memberikan sinyal peringatan tiap adanya perubahan didalam ruangan dan sistem keseluruhan. Sistem monitoring menampilkan tiap perubahan pada pembacaan PIR sensors reading dan kondisi perangkat lampu. Pemotongan jalur standar listrik dengan menghubungkan kabel listrik menggunakan relay – papan Raspberry Pi memberikan akses untuk otomasi proses saklar perangkat elektronik. Algoritme A-star memberikan alat penghitungan yang dibutuhkan sistem dalam menentukan waktu optimal untuk mematikan dan menghidupkan perangkat lampu dan elektronik lainnya.

#### Daftar Pustaka

- [1] G. Wibisono and A. Bayhaki, "Design and implementation of smart wireless street lighting system with ad-hoc network configuration," in *ELECO 2015 - 9th International Conference on Electrical and Electronics Engineering*, 2016.
  - [2] C. Leech, Y. P. Raykov, E. Ozer, and G. V. Merrett, "Real-time room occupancy estimation with Bayesian machine learning using a single PIR sensor and microcontroller," in *SAS 2017 - 2017 IEEE Sensors Applications Symposium, Proceedings*, 2017.
  - [3] C. Twumasi, K. A. Dotche, W. Banuenumah, and F. Sekyere, "Energy saving system using a PIR sensor for classroom monitoring," in *Proceedings - 2017 IEEE PES-IAS PowerAfrica Conference: Harnessing Energy, Information and Communications Technology (ICT) for Affordable Electrification of Africa, PowerAfrica 2017*, 2017.
  - [4] S. Lee, G. Tewolde, and J. Kwon, "Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application," in *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*, 2014.
  - [5] N. L. Ramli, N. Mohd Yamin, S. Ab Ghani, N. M. Saad, and S. A. Md Sharif, "Implementation of passive infrared sensor in street lighting automation system," *ARNP J. Eng. Appl. Sci.*, 2015.
  - [6] A. L. System, "PIR motion sensor," *US Pat. 7,579,595*, 2009.
  - [7] V. Vujovic, M. Maksimović, V. Vujović, N. Davidović, V. Milošević, and B. Perišić, "Raspberry Pi as Internet of Things hardware : Performances and Constraints Raspberry Pi as Internet of Things hardware : Performances and Constraints," *Des. Issues*, 2014.
  - [8] A. Bradbury and B. Everard, "Learning Python with Raspberry Pi," *Electronics*, 2014.
  - [9] R. C. Mason, "Fundamental relay-operating principles and characteristics," *Art Sci. Prot. Relaying*, 2015.
  - [10] L. H. S. Lelis, R. Stern, A. Felner, S. Zilles, and R. C. Holte, "Predicting optimal solution cost with conditional probabilities: Predicting optimal solution cost," *Ann. Math. Artif. Intell.*, 2014.
-