Implementasi *Emergency Light* untuk Jalur Evakuasi Terdekat Saat Gempa Menggunakan Algoritma *Greedy Search*

Dimas Rizki Agusta[1], Sidik Prabowo [2], Novian Anggis Suwastika[3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1] dimski@student.telkomuniversity.ac.id, [2]pakwowo@telkomuniversity.ac.id,
[3]anggis@telkomuniversity.ac.id

Abstrak

Letak geografis negara Indonesia masuk kedalam daerah *Ring of Fire* yang menyebabkan banyak terjadinya bencana gunung meleteus dan gempa bumi. Bencana gempa bumi juga banyak terjadi di Indonesia salah satu penyebabnya yaitu Indonesia mempunyai banyak gunung berapi yang aktif dan dapat menyebabkan gempa vulkanik. Gempa bumi juga bisa bisa disebabkan oleh beberapa faktor yaitu salah satunya akibat pergeseran lempeng tektonik yang bisa berakibat fatal pada kehidupan makhluk hidup bila terjadi dalam skala besar. Dengan kemajuan teknologi pada saat ini, sudah banyak teknologi yang diciptakan salah satunya untuk peringatan dini akan terjadinya gempa bumi hingga kecerdasan buatan yang dapat memprediksi kapan dan dimana gempa susulan itu akan terjadi. Untuk membantu dalam proses evakuasi dalam gedung secara efisien maka penilitian ini dibuat bertujuan untuk dapat menunjukan *emergency exit* atau pintu keluar terdekat dan jauh dari sumber getaran yang terindikasi lebih besar. Dalam sistem yang dibangun, proses validasi nantinya akan dibuktikan dengan cara membandingkan *final path* yang dihasilkan dengan hasil tiap *local optimum* yang dipilih. Pengujian dilakukan dengan dua skenario yang berbeda. Yang pertama yaitu terdapat indikasi anomali getaran pada *node* exit. Untuk skenario yang kedua yaitu tidak terdapat indikasi anomali getaran. Pada pengujian ini menghasilkan hasil akhir *final path* dengan tingkat akurasi 100% dengan hasil yang didapat tiap pengujian skenario berbeda tergantung dari skenario yang diujikan.

Kata kunci : Gempa bumi, *Emergency Light, Algoritma Greedy Search*

Abstract

The geographical location of Indonesia is included in the Ring of Fire area which causes many volcanic eruptions and earthquakes. Earthquake is also a lot happening in Indonesia, one of the causes is that Indonesia has many active volcanoes and can cause volcanic earthquake. Earthquake can also be caused by several factors, one of which is the shift of tectonic plates that can be fatal to the life of living things when occurring on a large scale. With technological advances at this time, there has been a lot of technology created; one of which is for early warning of earthquakes to artificial intelligence that can predict when and where the aftershocks will occur. To assist in the evacuation process in the buildings efficiently, the study is intended to be able to show the nearest emergency exit or exit and away from the source of vibration. In the built-in system, the validation process will later be proven by comparing the final path generated with the result of each selected local optimum. Testing was conducted with two different scenarios. The first scenario showed that there was an indication of vibration anomaly on the exit node. Meanwhile, for the second scenario, there is no vibration anomaly. The test results in a final path ending with an accuracy rate of 100% with the results obtained by each test scenario different depending on the tested scenario.

Keywords: Earthquake, Emergency Light, Algoritma Greedy Search

1. Pendahuluan

Latar Belakang

Bencana alam merupakan suatu peristiwa besar diluar kendali manusia. Berbagai bencana alam dapat terjadi di Indonesia. Sedangkan bencana alam yang paling sering terjadi di Indonesia adalah gempa bumi. Gempa bumi merupakan getaran yang terjadi di bumi dengan sebab tertentu. Gempa bumi terjadi bisa disebabkan oleh gerak lempeng bumi yang saling mendekat atau menjauh dan penyebab lainnya. Ketika terjadi gempa bumi, hal yang pertama dilakukan adalah penyelamatan diri yaitu dengan evakuasi diri melalui jalur evakuasi terdekat. Penentuan jalur evakuasi terefektif dapat dilihat dari tempat terdekat korban dengan pintu darurat.

Dengan memakai konsep *IoT* yaitu dapat menghubungkan semua perangkat ke internet dan memungkinkan setiap perangkat dapat berkomunikasi satu sama lainnya melalui internet [1]. Oleh karena itu, penulis berkesimpulan untuk membuat sistem *Emergency Light* dengan menggunakan algoritma *Greedy Search* dalam proses *Decision Making* untuk menentukan jalur terdekat. Penggunaan algoritma *Greedy Search* dalam penilitian ini dinilai cocok bagi penulis karena *node heuristic* yang dimunculkan dalam penilitian ini masih dalam jumlah yang sedikit sehingga dapat mempercepat proses komputasi untuk menentukan jalur evakuasi terdekat. Dalam sistem yang dibangun, penulis membuat sebuah maket berupa gedung mall dan membangkitkan beberapa *node heuristic* yang nantinya akan digunakan dalam penilitian ini. Pada saat gempa bumi terjadi, dibutuhkan proses evakuasi dengan cepat. Proses evakuasi yang cepat dapat dilakukan dengan mengimplementasikan algoritma *Greedy Search* dalam pemilihan sebuah jalur. Oleh karena itu, algoritma ini digunakan untuk proses *Decision Making* karena menggunakan pendekatan penyelesaian masalah dengan mencari *local maximum* atau *best solution* yang berdampak pada proses *Decision Making* [2].

Topik dan Batasannya

Adapun perumusan masalah yang didapat berdasarkan latar belakang masalah yaitu, bagaimana cara membangun sistem ini dengan mengimplementasikan algoritma *Greedy Search* dalam proses *Decision Making*. Penelitian ini memiliki batasan penelitian sebagai berikut :

1. Skenario pengujian dilakukan pada sebuah maket.
2. Sistem yang dibangun merupakan *prototype*.
3. Asumsi alat selalu memiliki daya listrik.
4. Tujuan penelitian ini hanya berfokus pada proses *Decision Making* dalam menentukan jalur evakuasi terdekat dan alatnya.

Tujuan

Tujuan yang ingin dicapai dalam Tugas Akhir (TA) ini adalah membangun sebuah sistem *Emergency Light* pada saat gempa; untuk menunjukkan jalur evakuasi terdekat dengan memanfaatkan konsep *IoT* dan dengan mengaplikasikan algoritma *Greedy Search* dalam proses *Decision Making*; hasil jalur evakuasi yang didapat sesuai berdasarkan jarak antar *node* yang telah diketahui pada penelitian ini.

Organisasi Tulisan

Penelitian TA ini disusun dengan struktur sebagai berikut : Bagian pertama yaitu pendahuluan yang berisi mengenai latar belakang, topik dan batasan, dan tujuan serta organisasi penulisan. Bagian kedua membahas tentang studi terkait dan referensi-referensi yang berkaitan dengan TA penulis. Bagian ketiga berisi mengenai penjelasan sistem yang dibangun. Bagian keempat berisi analisis dan evaluasi penelitian. Bagian kelima mengenai kesimpulan dari penelitian yang dilakukan.

2. Studi Terkait

Bab ini membahas tentang konsep dasar dan istilah yang dipakai dalam TA ini. Sub-bab yang pertama berisi tentang gempa bumi. Sub-bab yang kedua akan berisikan tentang definisi *LED*. Protokol komunikasi dalam sistem ini akan dijelaskan dalam sub-bab ketiga. Platform broker yang dipakai akan dijelaskan dalam sub-bab keempat. *Raspberry* Pi yang dipakai dalam penelitian ini akan dijelaskan dalam sub-bab kelima. Bab ini diakhiri dengan pembahasan tentang algoritma yang dipakai dan merupakan sebuah metode utama dalam TA ini.

Gempa Bumi

Gempa bumi yaitu salah satu bencana alam yang di sebabkan oleh pergeseran lapisan batuan di dasar permukaan bumi. Bencana gempa bumi dapat menyebabkan bencana lain, seperti tsunami jika terjadi pergerakan lempeng yang berasal di dasar laut. Gempa bumi dapat diklasifikasikan berdasarkan penyebabnya : Gempa Bumi Vulkanik, Gempa Bumi Tektonik, dan Gempa Bumi Terban [3]. Gempa bumi mempunyai bermacam-macam besaran gempa yang dapat dikategorikan, salah satunya Skala *Richter* (SR). Skala Richter atau SR didefinisikan sebagai logaritma (basis 10) dari amplitudo maksimum, yang diukur dalam satuan mikrometer, dari rekaman gempa oleh instrumen pengukur gempa (seismometer) Wood-Anderson, pada jarak 100 km dari pusat gempanya [4].

Light Emitting Diode (LED)

LED atau bisa disebut juga Light Emitting Diode merupakan komponen eleltronika yang dapat memancarkan beberapa warna seperti merah, hijau, putih, dll. LED merupakan bagian dari diode yang terbuat dari bahan semikonduktor. Warna dapat tercipta berdasarkan jenis bahan semikonduktor yang digunakannya [9]. LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata [9], selain itu LED juga biasanya digunakan sebagai komponen untuk menenjukan indicator terhadap sesuatu.

MQTT

Protokol *MQTT (Message Queuing Telemetry Transport)* adalah protokol yang berjalan diatas stack *TCP/IP* dan mempunyai ukuran paket data dengan *low overhead* yang kecil. Protokol ini termasuk jenis protokol *data-agnostic* yang artinya dapat mengirimkan data apapun seperti, data *binary*, *text*, dan *XML*. Stack *TCP/IP* juga sudah banyak didukung oleh banyak mikrokontroller seperti ESP8266 *WiFi SoC*, *Arduino*, *Raspberry Pi*, dll. Dalam sistem *MQTT* juga membutuhkan dua komponen perangkat lunak utama, yaitu: *MQTT Client*, dan *MQTT Broker* [10].

Mosquitto Platform

Mosquitto merupakan salah satu broker open source yang bekerja sebagai broker MQTT pada versi 3.1 dan 3.11. Mosquitto sangat mudah digunakan untuk semua perangkat dari single board komputer berdaya rendah sampai untuk sebuah server [11].

Raspberry Pi3

Raspberry pi merupakan mikrokontroller yang berfungsi sebagai *broker* atau media akses data. Raspi atau *Raspberry pi* dikembangkan oleh yayasan nirlaba, *Raspberry Pi Foundation*. Pada *Raspberry pi3 model B+* ini terdapat beberapa kelebihan dibanding dengan model sebelumnya, seperti : sudah menggunakan *chipset* baru yaitu *Broadcom* BCM2873B0 *Cortex* A53 64-bot 1,4GHz , sudah mendukung untuk koneksi *wireless 802.11ac* dan *Bluetooth* 4.2, dll [12].

Algoritma

Algoritma adalah suatu urutan dari beberapa langkah logis dan sistematis yang digunakan untuk menyelesaikan masalah tertentu [13]. Algoritma bisa banyak ditemukan dibanyak bahasa pemrograman salah satunya: *Python*. Salah satu jenis algoritma yang digunakan dalam penilitian ini adalah Algoritma *Greedy Search*. Algoritma *greedy* search merupakan jenis algoritma yang menggunakan pendekatan penyelesaian masalah dengan mencari *local optimum* dan berharap di dapatkannya *global optimum* di solusi akhir. Dalam persoalan optimasi ada dua macam, yaitu: maksimasi (maximization) dan minimasi (minimization) [14]. Pada proses implementasi algoritma tersebut kita terlebih dahulu menentukan *node* dan membuat *directed graph*. Proses selanjutnya yaitu kunjungi satu titik pada *graph*, dan ambil seluruh titik yang dapat dikunjungi dari titik sekarang. Setelah itu cari *local maximum* atau *minimum* ke titik selanjutnya dan tandai *graph* sekarang yang telah dikunjungi, dan pindah ke *local maximum* atau *minimum* yang telah ditentukan. Proses ini di lakukan berulang sampai titik tujuan didapatkan [2]. Pada algoritma *greedy search* memiliki

karakteristik proses komputasi yang cepat. Disamping kelebihan tersebut, terdapat kelemahan seperti hasil yang dihasilkan tidak selalu menjadi *best solution* atau solusi terbaik.

Penelitian lain pada subjek yang sama telah dilakukan oleh beberapa sarjana. Setiap penelitian akan dibahas sebagai berikut:

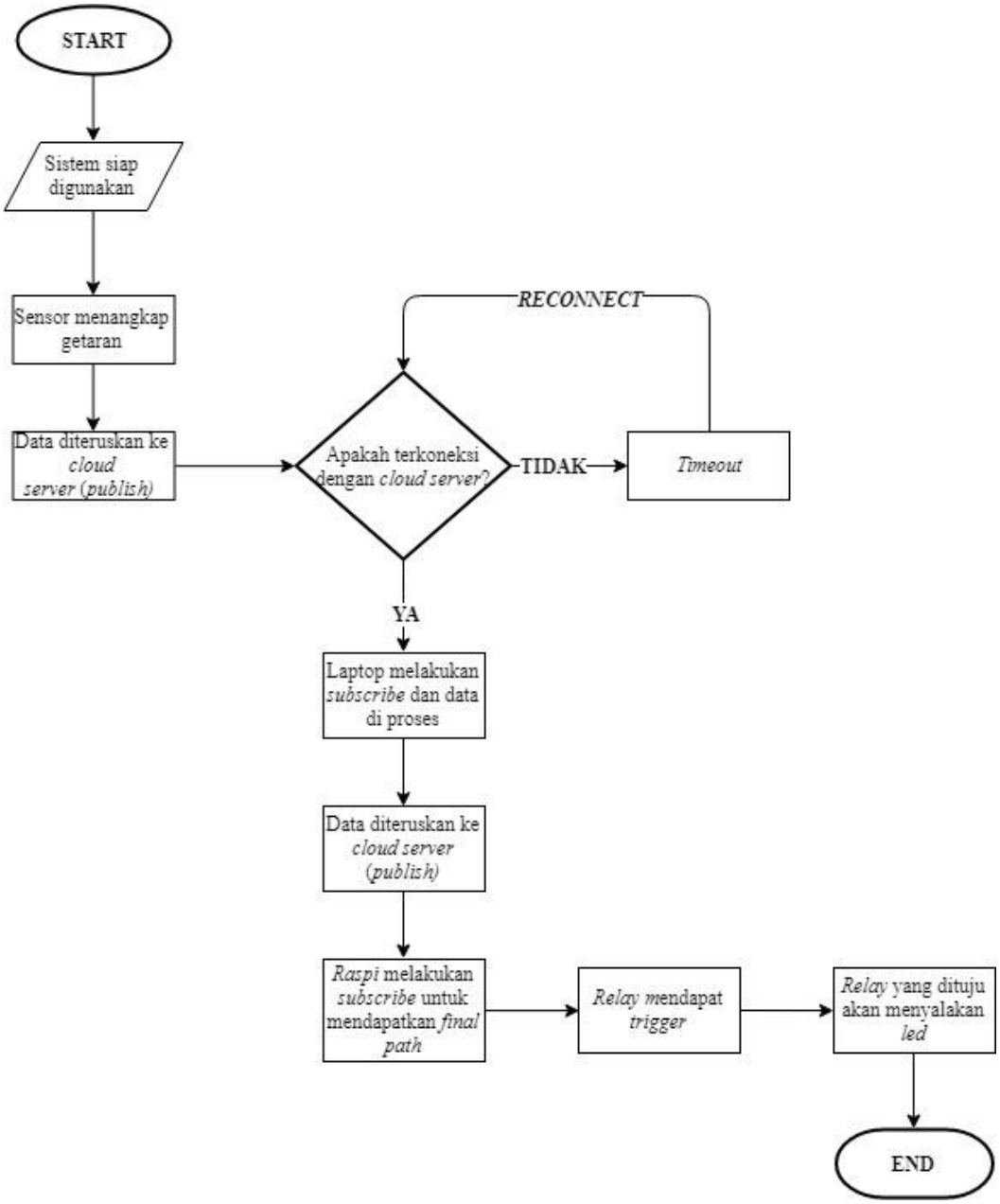
Kenta et.al (2017) pada penilitiannya, menciptakan sistem *Evacuation Guidance Support* atau *agent-based IoT (AIoT).* Dalam sistemnya terbagi menjadi dua yaitu *Function of Situation Recognition* yang berfungsi untuk mengetahui lokasi bencana dan informasi sensor. Pada penilitian ini penulis [7] menggunakan *device agent* dan *IoT devices* untuk membantu mengenali situasi bencana yang terjadi. Pada sistem yang kedua yaitu, *Function of Evacuation Guidance Planning by Agent*. Dalam sistem yang kedua ini, bertujuan untuk memberikan *evacuation guidance* sesuai dari informasi yang didapat pada sistem yang pertama. Pada penilitian [7], penulis memanfaatkan *UAV* dan *UGV* sebagai *agent-based Iot (AIoT).* Namun, penulis [7] belum memperhitungkan jumlah dari *UAV* dan *UGV* yang di gunakan untuk membantu proses evakuasi. Tujuan memperhitungkan jumlah dari *agent* yang digunakan yaitu untuk dapat mengoptimalkan proses evakuasi dan tidak terjadi *over budget* dalam melakukan implementasinya.

Christian et.al (2009) dapat merancang sebuah *emergency response system* untuk memberikan jalur evakuasi terdekat pada saat kebakaran. Dalam sistem ini terdiri dari: beberapa sensor dan kamera yang bertujuan untuk mendeteksi, dan memantau. Pada skenario yang terdapat dalam penilitian [6] kamera dan sensor akan saling terintegrasi untuk mendapatkan informasi dari kebakaran tersebut, dan selanjutnya akan di proses oleh *CPU*. *Evacuation Route* didapatkan dengan memakai *simple-path finding algorithm*. Dalam proses evakuasi yang dilakukan, pada penilitian [6] *CPU* akan memperhitungkan dan menentukan *evacuation route*. Jika nantinya *evacuation route* yang didapat terlalu dekat dengan area yang terindikasi berbahaya, maka *CPU* akan menentukan *alternative path*. Dalam penentuan *evacuation route* penulis [6] juga mempertimbangkan kepadatan pengungsi di suatu titik sehingga pada saat proses evakuasi dapat meminimalkan masalah kemacetan, dan potensi kepanikan. Dalam penilitian [6], disamping kelebihannya yang dapat mempertimbangkan kepadatan pengungsi menggunakan kamera yang selanjutnya *CPU* akan memproses dan mendapatkan *alternative path,* terdapat sebuah kelamahan dari sistem tersebut yaitu bagaimana jika kamera yang bertujuan melakukan *monitoring* tersebut tertutup dengan asap yang berasal dari kebakaran. Penulis [6] belum menyiapkan atas skenario yang dapat terjadi tersebut.

Amjath et.al (2017) dalam penilitiannya dapat merancang *Disaster-LINK* yaitu sebuah sistem perangkat *IoT* yang dapat bertindak sebagai alarm dan sistem pemantauan selama bencana alam, yang dapat saling berkomunikasi dengan *WiFi*. Penulis [5] dapat membangun sebuah peringatan dini yang terkoneksi dengan internet *(WiFi)* dan dapat menyebarkan informasinya ke *device* lain seperti *smartphone* melalui suara, *flashing light*, *SMS* dan *E-mail* notifikasi alarm. Pada penilitian [5] menggunakan internet karena dianggap lebih cepat daripada gelombang *seismic* gempa bumi, dan banyak lebih cepat daripada banjir atau tsunami. Hal ini dapat memungkinkan pesan peringatan terkirim jauh sebelum musibah sebenarnya, untuk selanjutnya dapat memberikan waktu pengguna untuk mengambil tindakan darurat pencegahan. Dari penilitian tersebut terdapat sebuah kelemahan yaitu belum adanya protokol keamanan yang di implementasikan pada sistem yang di rancang oleh penulis [5]. Keamanan data juga harus di pertimbangkan dalam sebuah sistem berbasis internet agar tidak terjadi *false alarm* atau *hoax*.

Himadri et.al (2017) dalam papernya yang berjudul "*Disaster Management using Internet of Things"* [8] dapat merumuskan *Emergency Management Lifecyle.* Dalam perumusannya tersebut, penulis [8] mempunyai beberapa proses yaitu : identifikasi resiko, kesiapsiagaan bencana, respon untuk keadaan darurat, alokasi sumber daya, perencanaan reaksi, dan pemulihan bencana. Disamping kelebihan yang terdapat pada penilitian [8] yaitu dapat menerapkan prinsip *IoT* untuk membuat *emergency management lifecycle*, terdapat pula sebuah kekurangannya yaitu memerlukan aktual dataset yang besar.

3. Sistem yang Dibangun



Gambar 1. *Flowchart* Perancangan Sistem

Dalam sistem yang dibangun pada TA ini data getar yang dipakai yaitu data getar yang sudah disimulasikan dan sudah terdapat pada *code* untuk melakukan proses *decision making* dan mendapatkan *final path* atau jalur evakuasi terdekat. Adapun proses tahapan dari sistem yang dibangun ini antara lain, pertama, sensor menangkap besaran getaran gempa, kedua, laptop melakukan *subscribe topic* dan memproses menggunakan algoritma *Greedy Search*, ketiga, *Raspi* akan melakukan *subscribe topic* untuk mendapatkan *final path*, dan yang terakhir yaitu *relay* yang dituju akan ter-*trigger* dan menyalakan *led.* Sistem yang dibangun dapat diilustrasikan seperti dibawah ini:

Penjelasan Gambaran Sistem

Sensor menangkap besaran getaran gempa
Pada tahap ini data getar akan disimulasikan atau di asumsikan sudah terbaca dengan benar. Pada pengujian nantinya data getar ini sudah *manually set* dan menyatu dengan *code* untuk proses *decision making*.

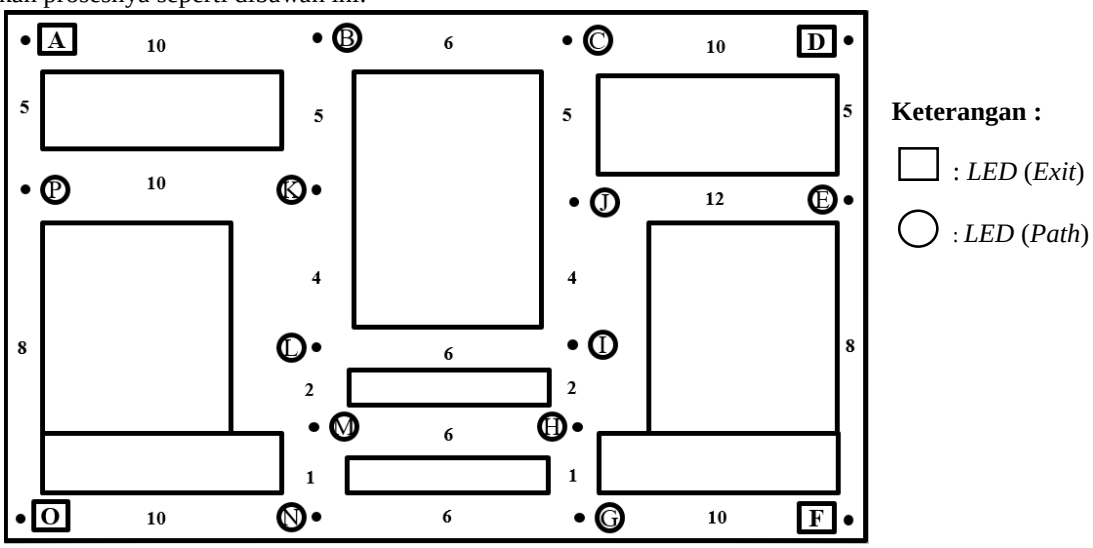
Laptop melakukan *subscribe topic* dan memproses data
Pada proses ini, Laptop akan langsung memproses data karena data getar sudah di simulasikan atau *manually set*. Data getar akan langsung di proses menggunakan algoritma *Greedy Search* dan mendapatkan jalur evakuasi terdekat *(final path).* Setelah mendapatkan data *final path* lalu data tersebut akan di publish ke broker yang berada di *cloud*.

Raspi* melakukan *subscribe topic
Pada proses ini *raspi* akan melakukan *subscribe topic* yang berisi data *final path.* Dari data yang didapat tersebut akan dilakukan *verifikasi* sebagai cara untuk mengaktifkan *GPIO* mana yang akan aktif dan mentrigger *relay.*

Relay* menyalakan *led
Setelah mendapat *trigger* dari *raspi*, *relay* lalu akan menyalakan *led* dan menunjukkan representasi dari *final path* atau jalur evakuasi terdekat.

Pengujian Skenario

Dalam pengujian sistem ini, nantinya akan di ujikan dalam 2 skenario, yaitu : skenario pertama memiliki 2 faktor yang akan mempengaruhi hasil dari *final path (initial position*, titik besar getaran yang di indikasi terjadi anomali), skenario kedua akan mencoba membuktikan apakah hasil atau *final path* yang didapat akan sama jika tidak terdapatnya indikasi anomali getaran. Pengujian sistem ini akan dilakukan disebuah maket dan akan dijelaskan prosesnya seperti dibawah ini:



Gambar 2. Maket Gedung Mall Lt.1

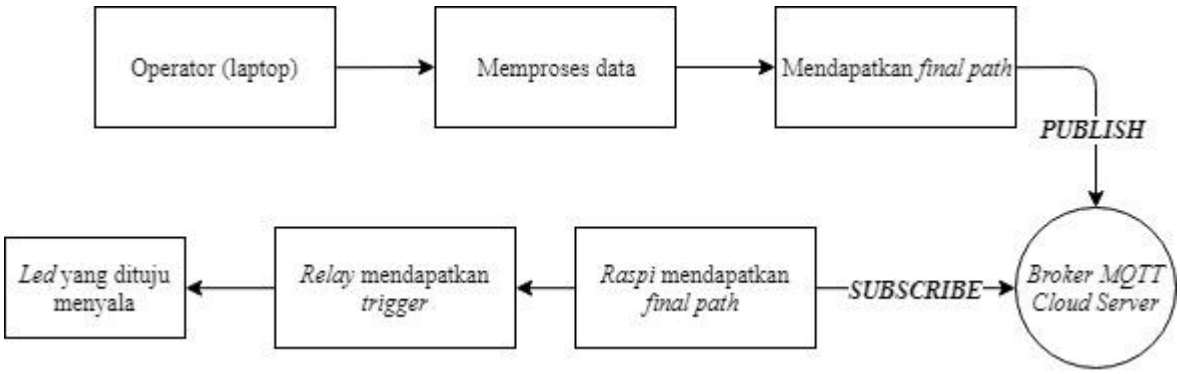
Pada skenario ini hal yang pertama dilakukan adalah membangkitkan *node heuristic.* Membangkitkan *node* tersebut adalah sebagai syarat untuk nantinya di proses oleh algoritma *Greedy Search*. Alasan membangkitkan titik *node* yaitu karena dalam skenario ini tujuannya adalah untuk mencari jalan evakuasi terdekat dengan memakai *algoritma greedy*, maka dibutuhkan sebuah *graph* atau *node* untuk menghitung jarak dari titik satu ke titik lainnya. *Node* tersebut nantinya akan dipakai untuk data pada tabel yang berisikan jarak antar titik *node* tersebut dan selanjutnya akan diseleksi menggunakan *algoritma greedy.*

Kemudian sensor akan diletakkan pada *node A, node D, node F,* dan *node O*. Selanjutnya yaitu peletakan *relay* yang nantinya menjadi *aktuator* untuk menyalakan *led* yang dipakai sebagai penunjuk jalur evakuasi.

Peletakan *relay* nantinya disesuaikan dengan titik *node* (selain *node exit*) untuk dapat menunjukan jalur evakuasi terdekat berdasarkan lokasi atau di titik korban berada.

Rancangan Sistem

Pada perancangan sistem yang akan digunakan dalam penilitian ini terdiri dari beberapa bagian, yaitu :, *Raspberry pi3*, *Cloud Server* (*Mosquitto)*, *Led*, dan laptop. Berikut rancangan arsitektur sistem :



Gambar 3. Arsitektur Sistem

Arsitektur sistem diatas akan dijelaskan lebih detil sebagai berikut:

Pada penilitian ini, nantinya data getar sudah di simulasikan atau di asumsikan sudah terbaca dengan baik. Proses selanjutnya yaitu laptop akan langsung memproses data getar dan akan menghasilkan *final path. Final path* yang didapat lalu dikirimkan ke *raspi* melalui *broker cloud server.* Setelah itu, *Raspberry pi* akan melakukan *subscribe* topic dan diproses mana *GPIO* atau *relay* yang akan aktif dan pada akhirnya mengaktifkan *led* untuk menunjukkan jalur evakuasi terdekat.

4. Evaluasi

4.1 Hasil Pengujian

Skenario pengujian pada penelitian ini akan dilakukan masing-masing 5 kali percobaan untuk dua skenario, yaitu untuk skenario pertama jika terindikasi terdapat anomali getaran dan skenario kedua jika tidak terindikasi anomali getaran. Pada pengujian ini akan membuktikan tingkat akurasi yang dihasilkan dengan cara membandingkan hasil *final path* yang di dapat dengan hasil tiap *local optimum* yang dipilih pada setiap iterasinya. Pada pengujian skenario ini terdapat kondisi jika *node* sudah berada dekat pada pintu keluar maka *node* tersebut akan dipilih. Dalam kolom iterasi dibawah ini nantinya akan menunjukkan hasil perbandingan antar *node* yang dihasilkan oleh algoritma *greedy search* dan dapat membuktikan hasil yang di dapat tiap iterasi. Untuk waktu eksekusi pada setiap pengujian skenario yang dilakukan adalah sama yaitu 0.1s.

- Pada skenario yang pertama ini, akan menguji tingkat akurasi jika diketahui **terdapat** anomali getaran pada ***node* = 'd'**. Didapatkan tingkat akurasi untuk skenario pertama yaitu 100%.

No	*Initial Position*	*Final Path*	Iterasi					
			1	2	3	4	5	Valid / Tidak Valid
1.	K	K,L,M,N,O	[B:5, L:4, P:10] = L	[I:6, K:199, M:2] = M	[H:6, L:199, N:2] = N	*Node exit* = O	-	Valid
2.	C	C,J.I.H.G.F	[B:6, D:167, J:5] = J	[C:199, E:10, I:4] = I	[J:199, H:2, L:6] = H	[I:199, M:6, G:1] = G	*Node exit* = F	Valid
3.	B	B,A	*Node exit* = A	-	-	-	-	Valid
4.	A	A	*Node exit* = A	-	-	-	-	Valid
5.	E	E,F	*Node exit* = F	-	-	-	-	Valid

Tabel 1. Hasil Pengujian Skenario Pertama

- Pada skenario yang kedua ini, akan menguji tingkat akurasi jika diketahui **tidak** terdapat anomali getaran. Didapatkan tingkat akurasi untuk skenario kedua yaitu 100%.

No	*Initial Position*	*Final Path*	Iterasi					
			1	2	3	4	5	Valid / Tidak Valid
1.	K	K,L,M,N,O	[B:5, L:4, P:10] = L	[I:6, K:199, M:2] = M	[H:6, L:199, N:1] = N	*Node exit* = O	-	Valid
2.	C	C,D	*Node exit* = D	-	-	-	-	Valid
3.	B	B,A	*Node exit* = A	-	-	-	-	Valid
4.	A	A	*Node exit* = A	-	-	-	-	Valid
5.	E	E,D	*Node exit* = D	-	-	-	-	Valid

Tabel 2. Hasil Pengujian Skenario Kedua

4.2 Analisis Hasil Pengujian

Bedasarkan hasil pengujian diatas dapat dilihat bahwa tingkat akurasi dari kedua skenario tersebut yaitu 100%. Pada pengujian skenario pertama terdapat indikasi anomali getaran pada *node* = ‘d’ dan dalam percobaan nomor dua dan lima menunjukkan bahwa seberapapun dekatnya posisi korban dengan *node exit* maka node tersebut tidak akan terpilih karena terdapat anomali getaran pada titik itu. Oleh karena itu, jika *node exit* tersebut adalah node yang seharusnya tidak dilewati, maka sistem ini dapat bekerja dengan benar yaitu tetap tidak mengarahkan ke *node exit* yang terindikasi terdapat anomali getaran. Selanjutnya hasil pengujian skenario kedua didapatkan tingkat akurasi 100%. Pada pengujian skenario yang kedua tidak terdapat indikasi anomali getaran. *Initial Position* yang diujikan pada skenario kedua dibuat sama agar dapat membandingkan hasil *final path* jika tidak terdapat indikasi anomali getaran. Maka dapat disimpulkan bahwa, jika *node exit* tidak terdapat indikasi anomali getaran seperti pada skenario pengujian kedua pada nomor dua dan lima, maka node tersebut akan masuk menjadi pilihan dalam proses perbandingan antar *node.*

5. Kesimpulan

Sistem yang dirancang berhasil bekerja dengan baik dengan mengimplementasikan algoritma *greedy search* sebagai *AI* untuk proses penentuan jalur evakuasi terdekat. Hasil tingkat akurasi yang didapat dari dua pengujian skenario diatas yaitu 100%. Metode validasi yang digunakan yaitu dengan cara membandingkan *final path* yang dihasilkan dengan hasil tiap *local optimum* yang dipilih. Dari pengujian skenario tersebut dapat juga disimpulkan bahwa hasil *final path* yang dihasilkan didapat berdasarkan faktor yang mempengaruhinya, seperti pada *node exit* apakah terdapat indikasi anomali getaran atau tidak.