



RANCANG BANGUN SISTEM PENENTUAN JALUR EVAKUASI KELUAR GEDUNG RUMAH SAKIT SAAT KONDISI DARURAT MENGGUNAKAN ALGORITMA DIJKSTRA DENGAN ANTRIAN PRIORITAS

Ade Putera Kemala¹, Dody Ichwana²

Jurusan Sistem Komputer FTI Universitas Andalas Limau Manis Kota Padang 25163 INDONESIA

ARTICLE INFORMATION

Received: February 18, 2019
 Revised: March 8, 2019
 Available online: March 29, 2019

KEYWORDS

Sistem evakuasi, algoritma djikstra dengan antrian prioritas.

CORRESPONDENCE

Phone:
 E-mail: dody.ichwana@fti.unand.ac.id

A B S T R A C T

Disaster is a sudden event, such as an accident or a natural catastrophe, that causes great damage or loss of life. Disasters can occur at any time, therefore a mechanism is needed to evacuate out of the building during a disaster. Generally public facilities have provided instructions and evacuation routes outside the building. Based on the severity of the disaster and the evacuation capability of the victim, the evacuation strategy can range from evacuation as soon as possible, evacuate slowly, move to a safe location inside the building or take refuge in the available protection room and wait for the rescue team to arrive. The algorithm for finding the shortest paths can be used to determine the evacuation route. But this path is still static, if the route damaged in then the evacuation route would become useless. The time for the evacuation process can also increase if the condition of the evacuation route is not known whether damaged or not. The solution to solve this problem is to make a system that can help find the safest and shortest evacuation routes during emergencies. This system consist of microcontroller Arduino Mega to control the system and led for evacuation sign. The evacuation routes is determined by implementing dijkstra algorithm with priority queue to search the shortest path.

PENDAHULUAN

Bencana merupakan peristiwa atau rangkaian peristiwa yang mengancam dan mengganggu kehidupan dan penghidupan masyarakat yang disebabkan, baik oleh faktor alam dan/atau faktor nonalam maupun faktor manusia sehingga mengakibatkan timbulnya korban jiwa manusia, kerusakan lingkungan, kerugian harta benda, dan dampak psikologis[1]. Bencana dapat terjadi kapan saja, oleh karena itu dibutuhkan suatu mekanisme untuk evakuasi keluar gedung pada saat terjadinya bencana. Umumnya fasilitas umum sudah menyediakan petunjuk dan rute evakuasi keluar gedung. Berdasarkan tingkat parahnya bencana dan kemampuan evakuasi korban, strategi evakuasi dapat berubah mulai dari evakuasi secepatnya, evakuasi dengan lambat, berpindah ke lokasi yang aman di dalam gedung atau berlindung di ruangan perlindungan yang tersedia dan menunggu kedatangan tim penyelamat[2].

Gedung rumah sakit selalu memiliki penghuni selama 24 jam baik itu pekerja, pasien atau pengunjung. Sebagian penghuni merupakan pasien yang kondisi kesehatannya tidak optimal sehingga menyulitkan dalam mencari jalur evakuasi tanpa adanya bantuan. Selain itu umumnya rumah sakit merupakan bangunan berbentuk gedung, sehingga sesuai dengan tempat implementasi sistem yang dikembangkan ini. Berdasarkan keputusan menteri kesehatan[3], rumah sakit di Indonesia harus memiliki protokol evakuasi dan tim penyelamat pada saat kondisi darurat. Protokol ini terbagi atas berbagai kode warna, contohnya kode biru menandakan adanya pasien yang mengalami serangan jantung dan butuh pertolongan. Protokol untuk keadaan gempa adalah kode hijau, saat protokol ini dijalankan petugas rumah sakit akan membantu pasien untuk evakuasi keluar gedung. Kode merah untuk bencana kebakaran, kode hitam digunakan pada saat adanya ancaman bom.

Berdasarkan keputusan menteri kesehatan[4], rumah sakit harus memiliki jalur dan rambu-rambu evakuasi keluar gedung saat kondisi darurat. Rumah sakit harus memiliki staff atau personil yang terlatih untuk menangani proses evakuasi saat kondisi darurat[5]. Algoritma pencari rute terpendek dapat digunakan untuk menentukan rute evakuasi[6]. Tetapi jalur tersebut masih bersifat statik, apabila gedung mengalami kerusakan di bagian jalur evakuasi tersebut maka jalur evakuasi tersebut menjadi tidak berguna lagi. Waktu untuk proses evakuasi dapat meningkat apabila kondisi rute evakuasi tidak diketahui rusak atau tidak.

Algoritma Dijkstra Dengan Antrian Prioritas

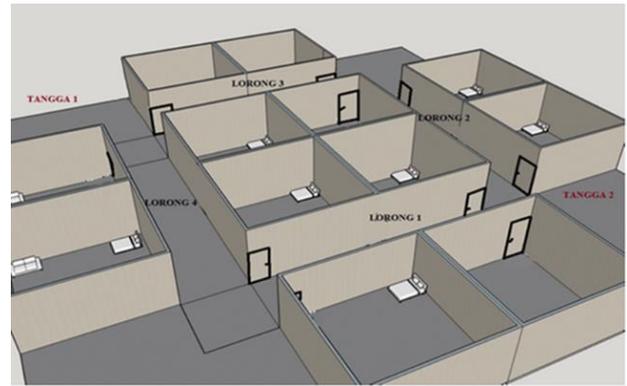
Algoritma dijkstra dengan antrian prioritas merupakan bentuk pengembangan dari algoritma dijkstra. Algoritma dijkstra ditemukan oleh peneliti bernama Edsger W. Dijkstra, dan dipublikasikan pada tahun 1959 [7]. Algoritma dijkstra menggunakan strategi greedy[8] yang bekerja dengan cara membandingkan nilai terpendek menuju simpul yang terhubung. Simpul yang memiliki nilai terkecil akan dipilih dan simpul dengan nilai yang lebih besar akan diabaikan dan tidak akan diproses untuk seterusnya. Kekurangan dari strategi ini adalah jalur yang ditemukan bisa jadi bukan rute terpendek. Hal ini terjadi karena simpul-simpul yang nilainya besar akan diabaikan walaupun ada kemungkinan simpul tersebut terhubung dengan jalur yang lebih pendek. Menurut F. Benjamin Zhan [9] kekurangan ini dapat diatasi dengan mengimplementasi struktur data pada algoritma dijkstra, seperti struktur data antrian prioritas. Algoritma dijkstra dengan antrian prioritas akan mencari jalur terpendek antara simpul pada suatu graf. Simpul dengan nilai terkecil akan dipilih untuk diproses, sedangkan simpul yang nilainya besar akan disimpan sementara di dalam antrian prioritas. Dengan pemanfaatan struktur data, algoritma dapat bekerja mencari rute yang lebih optimal dari segi jarak. Pada sistem ini, graf yang digunakan merupakan pemodelan dari denah rancangan prototipe yang akan dibuat. Pengguna teknologi diharapkan menjadi solusi memudahkan pekerjaan [10] [11].

METODOLOGI PENELITIAN

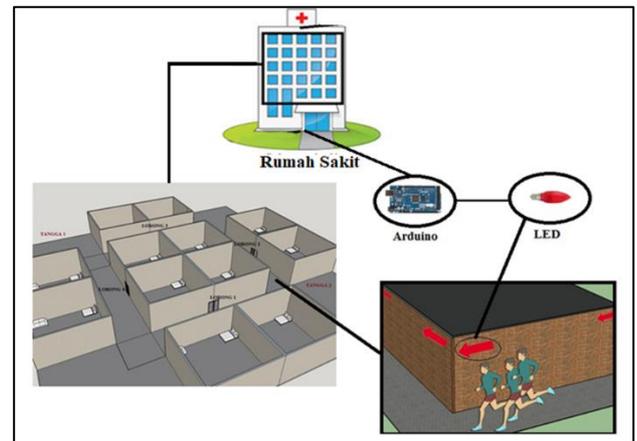
Jenis penelitian yang digunakan dalam penulisan tugas akhir ini adalah penelitian eksperimental (*Experimental Research*). Penelitian eksperimental adalah jenis penelitian yang digunakan untuk melihat hubungan sebab akibat. Penelitian eksperimental digunakan untuk mencari pengaruh perlakuan terhadap yang lain dalam kondisi yang terkontrol. Rancangan penelitian dibutuhkan sebagai dasar dalam melakukan penelitian demi mencapai tujuan yang telah ditetapkan. Penelitian ini bertujuan untuk membuat suatu sistem yang dapat menentukan jalur evakuasi keluar gedung rumah sakit saat kondisi darurat.

Rancangan Umum Sistem

Rancangan umum sistem terdiri dari beberapa komponen hardware seperti Arduino Mega dan LED sebagai penunjuk rute evakuasi. Arduino Mega berfungsi sebagai pengendali sistem. LED digunakan sebagai penunjuk rute keluar gedung. Bentuk rancangan umum dari sistem evakuasi keluar gedung saat bencana gempa dapat dilihat pada Gambar 1 dan Gambar 2.



Gambar 1. Rancangan Bagian Dalam Gedung Lantai 1 dan Lantai 2



Gambar 2. Rancangan Umum Sistem

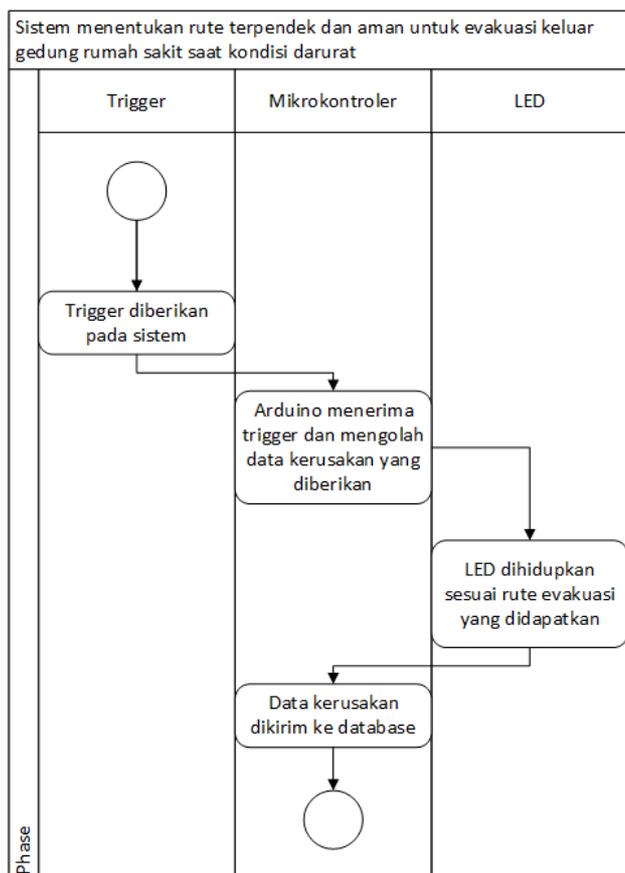
Pada perancangan sistem ini dibangun prototipe gedung rumah sakit yang memiliki 2 lantai. Cara kerja sistem ini adalah, saat bencana terjadi dan trigger bencana diberikan, sistem akan aktif. Setelah sistem aktif mikrokontroler akan mengolah data kerusakan dummy yang diberikan. Rute terpendek dan aman akan ditentukan oleh mikrokontroler menggunakan algoritma dijkstra dengan antrian prioritas berdasarkan data kerusakan yang diberikan. Setelah rute evakuasi didapatkan, LED akan dihidupkan sebagai penunjuk jalan berdasarkan hasil dari perhitungan oleh mikrokontroler. Data kerusakan dikirim ke database oleh mikrokontroler untuk digunakan oleh sistem penyelamatan pasien.

Rancangan Proses

Sistem evakuasi keluar gedung rumah sakit saat bencana ini terdiri dari dua bagian utama, yaitu hardware dan software. Perancangan sistem ini bertujuan agar alur kerja sistem dapat dipahami, mulai dari pembacaan data sensor hingga menampilkan data untuk membantu tim evakuasi. Proses yang dijalankan oleh sistem ini dapat dilihat pada Gambar 3.

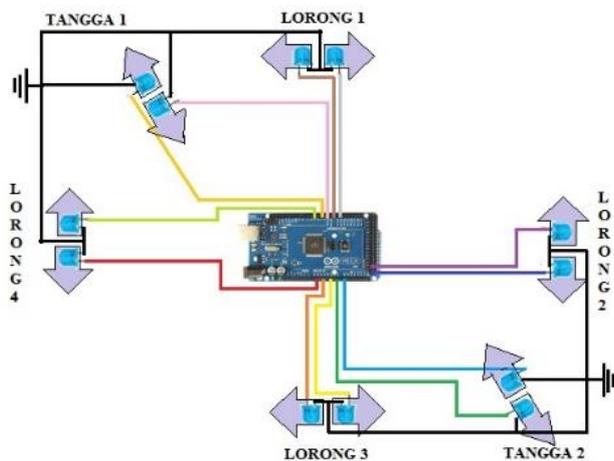
Rancangan Perangkat Keras

Pembuatan sistem evakuasi pasien rumah sakit saat bencana ini adalah berupa prototipe gedung 2 tingkat yang terbuat dari kayu dan triplek serta menggunakan arduino mega sebagai pengendali sistemnya. Prototipe gedung untuk sistem ini dirancang memiliki 12 kamar per lantai. Lantai dihubungkan oleh 2 tangga dan setiap lantai memiliki 4 lorong yang menghubungkan tiap kamar.



Gambar 3. Activity Diagram

Gambar 4 menampilkan rancangan perangkat keras dari sistem. Pengendali sistem merupakan Arduino. Pada saat sistem diberikan trigger, arduino akan bekerja mengolah data dummy yang diberikan untuk menghitung jalur terpendek. Setelah rute didapatkan, led penunjuk arah akan dihidupkan. Semua perangkat dihubungkan langsung ke Arduino menggunakan jumper atau kabel.

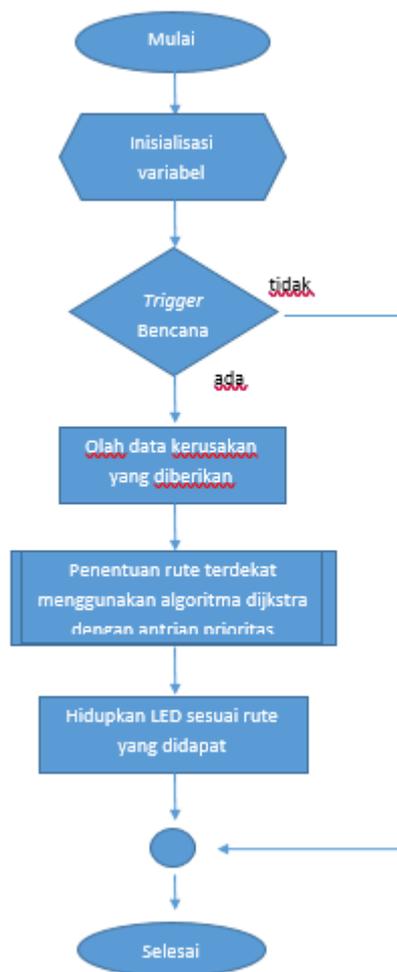


Gambar 4. Rancangan perangkat keras

Rancangan Perangkat Lunak

Perangkat lunak yang dirancang pada sistem evakuasi keluar gedung rumah sakit terbagi atas 2 bagian. Perancangan yang pertama adalah aplikasi desktop yang berfungsi untuk memberikan data dummy ke perangkat Arduino. Perancangan

yang kedua adalah aplikasi tertanam pada mikrokontroler yang berfungsi sebagai pengendali keseluruhan sistem. Cara kerja perangkat lunak aplikasi tertanam secara umum dapat dilihat di flowchart pada Gambar 5.



Gambar 5. Flowchart Perangkat Lunak

Algoritma Dijkstra Dengan Antrian Prioritas

Sistem ini menggunakan algoritma dijkstra dengan antrian prioritas untuk mencari rute terpendek menuju pintu keluar. Algoritma dijkstra dengan antrian prioritas merupakan algoritma yang berfungsi untuk mencari rute terpendek pada suatu graf. Langkah-langkah cara kerja algoritma Dijkstra dengan antrian prioritas dapat dilihat pada list berikut:

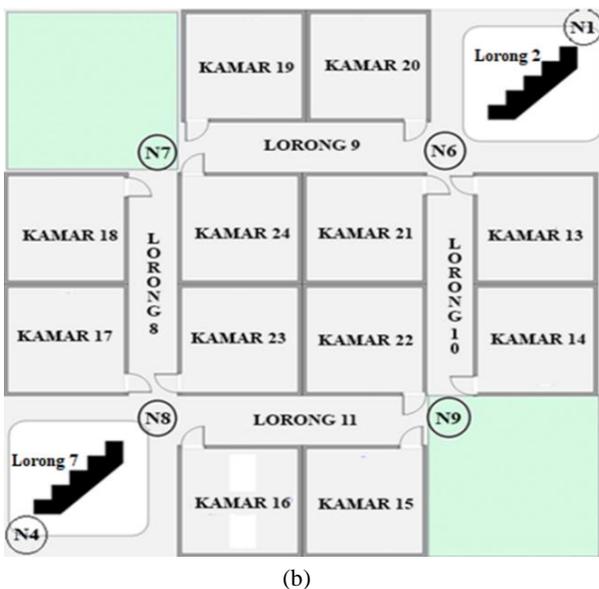
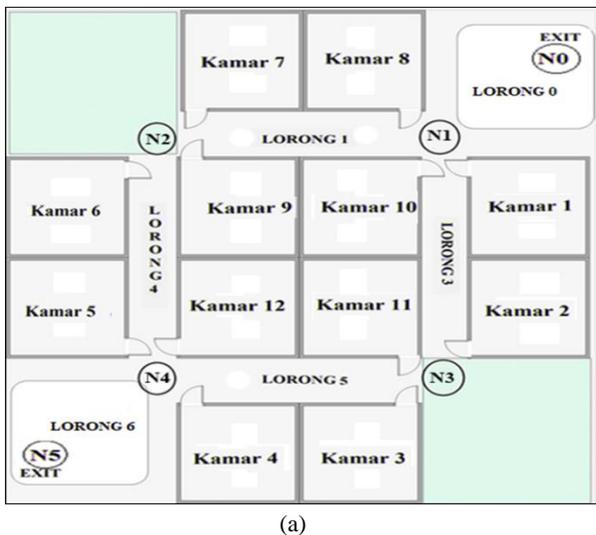
1. Algoritma dimulai
2. Deklarasikan variable yang dibutuhkan
3. Membentuk graf sesuai data kerusakan yang diberikan
4. Masukkan nilai sementara pada variable yang menyimpan nilai jarak
5. Masukkan data simpul awal kedalam antrian prioritas
6. Lakukan perulangan selama antrian prioritas tidak kosong
7. Ambil simpul (u) dengan nilai terkecil yang ada di dalam antrian prioritas
8. Lakukan perulangan selama ada simpul yang terhubung dengan simpul (u) yang diambil dari antrian prioritas
9. Ambil data simpul yang terhubung dengan simpul (u), simpan dalam simpul (x)
10. Jika jarak simpul (x) > jarak simpul (u) + jarak simpul (x)

11. Ganti nilai jarak simpul (x) dengan jarak simpul (u) + jarak simpul (x)
12. Simpan rute menuju simpul (x) dengan nilai simpul (u)
13. Simpan data simpul (x) ke dalam antrian prioritas
14. Selesai.

Algoritma dijkstra memiliki kompleksitas waktu sebesar $O(E \log V)$. E merupakan total jumlah sisi dan V merupakan total jumlah node pada graf. Jika dipecah $O(E \log V)$ akan menjadi $O(n) \times O(\log V)$. Kompleksitas $O(n)$ artinya waktu proses algoritma akan meningkat sebanding dengan meningkatnya jumlah input (n). $O(\log V)$ artinya waktu proses algoritma akan meningkat seiring meningkatnya jumlah input(V) secara eksponensial. Gambar 6 menampilkan denah rancangan lantai yang akan digunakan. Rancangan prototipe sistem yang akan dibangun memiliki 2 lantai. Lantai 1 dan lantai 2 memiliki denah yang sama yang dihubungkan dengan tangga. Setiap sudut persimpangan lorong ditandai sebagai simpul graf. Tangga darurat pada lantai dua menuju lantai satu dan langsung terhubung ke pintu darurat keluar gedung. Berdasarkan Gambar 7 didapatkan data hubungan lorong dengan simpul yang dapat dilihat pada Tabel 1.

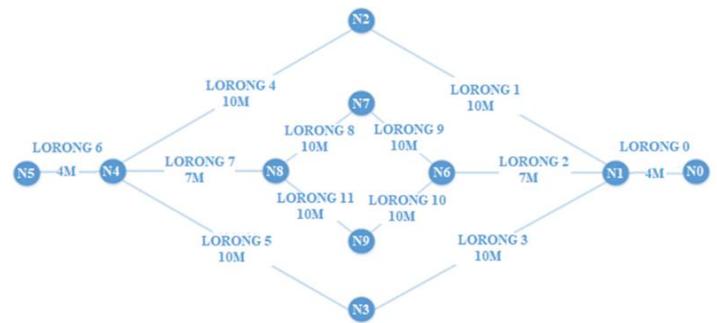
Tabel 1. Hubungan Lorong Dengan Simpul

Lorong	Simpul yang Terhubung	Panjang Lorong
Lorong 0	Simpul 0 Simpul 1	4
Lorong 1	Simpul 1 Simpul 2	10
Lorong 3	Simpul 1 Simpul 3	10
Lorong 4	Simpul 2 Simpul 4	10
Lorong 5	Simpul 3 Simpul 4	10
Lorong 6	Simpul 4 Simpul 5	4
Lorong 2	Simpul 1 Simpul 6	7
Lorong 7	Simpul 4 Simpul 8	7
Lorong 8	Simpul 7 Simpul 8	10
Lorong 9	Simpul 6 Simpul 7	10
Lorong 10	Simpul 6 Simpul 9	10
Lorong 11	Simpul 8 Simpul 9	10



Gambar 6. Denah Lantai (a)Denah Lantai 1; (b)Denah Lantai 2

Berdasarkan Gambar 6, dapat dibuatkan graf denah jalur evakuasi yang dapat dilihat pada Gambar 7.



Gambar 7. Graf Denah Gedung Rancangan

Graf pada Gambar 7 merupakan graf dimana semua kondisi lorong dan tangga pada rumah sakit dalam kondisi aman dan tidak rusak. N0 dan N5 merupakan pintu keluar, lorong 2 dan lorong 7 merupakan tangga yang menghubungkan lantai 1 dan lantai 2. Pada saat bencana atau kondisi darurat keadaan gedung bisa berubah. Data lorong atau tangga rusak yang digunakan pada sistem ini menggunakan data dummy. Berdasarkan data tersebut, akan dibentuk graf baru yang sesuai dengan kondisi gedung tersebut. Kemudian menggunakan algoritma dijkstra dengan antrian prioritas, mikrokontroler akan menghitung rute terpendek untuk keluar dari gedung. Hasil perhitungan akan ditampilkan dengan menhidupkan lampu led berbentuk tanda panah untuk menunjukkan jalur keluar.

Dalam penggunaannya, algoritma dijkstra dengan antrian prioritas dapat digunakan untuk mencari jalur terpendek antara dua simpul. Selain itu algoritma ini juga dapat dimanfaatkan untuk mencari jalur terpendek menuju semua simpul pada graf. Pada pembuatan sistem ini, algoritma dijkstra dengan antrian prioritas digunakan untuk mencari jalur terpendek dari simpul awal menuju semua simpul. Dalam hal ini, yang menjadi pintu keluar adalah simpul N0 dan N5. Algoritma dijkstra dengan antrian prioritas akan dijalankan dua kali, yang pertama mencari semua rute terpendek menuju semua simpul dengan pintu keluar

pertama (N0) sebagai simpul awalnya. Kemudian pada perhitungan yang kedua, pintu keluar kedua (N5) akan digunakan sebagai simpul awal. Hasil kedua perhitungan tersebut kemudian akan dibandingkan untuk mendapatkan jalur yang terpendek.

HASIL DAN PEMBAHASAN

Implementasi Perangkat Keras

Implementasi perangkat keras rancang bangun sistem pencari rute evakuasi terpendek keluar gedung ini dibangun menggunakan perangkat keras berupa mikrokontroler arduino mega dan led. Komponen rancangan perangkat keras yang diimplementasikan dapat dilihat pada Gambar 8.



Gambar 8. Implementasi Perangkat Keras

Implementasi Perangkat Lunak

Implementasi perangkat lunak pada perancangan sistem ini menggunakan pemrograman Arduino IDE dan visual studio dan terbagi menjadi dua bagian yaitu Aplikasi desktop yang dibangun menggunakan visual studio. Aplikasi ini digunakan oleh user untuk menentukan lorong yang rusak, dan mengirim data dummy tersebut ke Arduino mega lewat komunikasi serial. Total lorong pada rancangan sistem adalah 12 lorong dengan 2 macam kemungkinan yaitu rusak dan tidak rusak. Total kemungkinan skenario kerusakan yang dapat dibuat adalah 2^{12} atau 4096 kemungkinan. Tampilan aplikasi desktop dapat dilihat pada Gambar 9.

Aplikasi kedua adalah perangkat lunak yang tertanam di Arduino mega. Program pada Arduino mega akan menerima data kerusakan dari aplikasi desktop dan mengolah data tersebut menggunakan algoritma dijkstra dengan antrian prioritas untuk mencari rute terpendek. Setelah rute didapatkan, mikrokontroler akan menyalakan led sesuai rute yang telah dihitung.

Pengujian Sistem

Pengujian ini dilakukan untuk mengukur waktu yang diperlukan oleh perangkat Arduino Mega untuk mengolah data menggunakan algoritma dijkstra dengan antrian prioritas dan menyalakan led. Pengujian ini dilakukan 10 kali menggunakan skenario kerusakan yang berbeda. Skenario kerusakan yang digunakan untuk semua pengujian dapat dilihat pada Tabel 2. Data waktu hasil pengujian dapat dilihat pada Tabel 3.

Tabel 2. Skenario Kerusakan

Skenario Kerusakan	Lorong											
	0	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	1	0	0	0	0	1	1	0	0
2	1	0	0	0	1	1	0	0	0	0	1	0
3	0	1	0	0	0	0	1	1	0	1	0	1
4	0	0	0	0	0	0	0	0	1	1	1	1
5	1	0	0	0	0	0	1	0	0	0	0	0
6	0	0	0	0	1	1	1	0	1	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	1	0	0	0	1	0	1	0	0	0
9	0	0	1	0	0	0	0	1	0	0	0	0
10	1	1	0	0	0	1	0	0	0	1	0	1

Berdasarkan Tabel 2, ada 10 skenario kerusakan yang akan digunakan pada pengujian. Lorong yang nilainya adalah 0 dalam kondisi baik dan lorong yang nilainya 1 dalam kondisi rusak dan tidak bisa dilalui.

Tabel 3. Pengujian Waktu Pengolahan Data Kerusakan

Skenario Kerusakan	Waktu	
	Algoritma Dijkstra	Menghidupkan Led
1	4 ms	1 ms
2	3 ms	0 ms
3	3 ms	0 ms
4	4 ms	0 ms
5	3 ms	0 ms
6	3 ms	0 ms
7	6 ms	0 ms
8	3 ms	1 ms
9	4 ms	0 ms
10	3 ms	0 ms
Rata-Rata	3.6 ms	0.2 ms
Standar Deviasi	0.9333 ms	0.1777 ms

Berdasarkan Tabel 3, pengujian dilakukan sebanyak 10 kali dengan 10 data kerusakan yang berbeda. Dari hasil pengujian didapatkan waktu tercepat untuk menghitung rute adalah 3 ms dan waktu terlama adalah 6 ms. Sedangkan untuk menyalakan led, waktu terlama adalah 1 ms sedangkan waktu tercepat adalah 0 ms atau kecil dari 1 ms sehingga tidak dapat dihitung oleh program. Waktu rata-rata untuk menjalankan algoritma dijkstra dengan antrian prioritas adalah 3.6 ms, sedangkan waktu rata-rata untuk menyalakan led adalah 0.2 ms. Nilai standar deviasi yang didapatkan untuk pengujian algoritma dijkstra dengan antrian prioritas adalah 0.9333 ms sedangkan untuk menyalakan led didapatkan nilai standar deviasi sebesar 0.1777 ms.



Gambar 9. Tampilan aplikasi desktop

Pengujian Aplikasi Desktop

Pengujian pada aplikasi desktop dilakukan untuk mengetahui apakah aplikasi desktop dapat mengirim data ke mikrokontroler dan memastikan bahwa data yang dikirim sama dengan data yang diterima mikrokontroler.

Data kerusakan yang digunakan pada pengujian ini sama dengan data yang ada pada tabel 3. Pengujian ini dilakukan sebanyak 10 kali. Dikarenakan pin rx/tx digunakan untuk mentransfer data dari aplikasi desktop ke mikrokontroler, maka data yang masuk tidak dapat dilihat melalui serial monitor. Oleh karena itu, program dimodifikasi dan dibuat agar data yang diterima ditampilkan pada perangkat lcd 16x2. Data tersebut kemudian dibandingkan apakah data yang dikirim sama dengan data yang diterima. Hasil pengujian ini dapat dilihat pada Tabel 4.

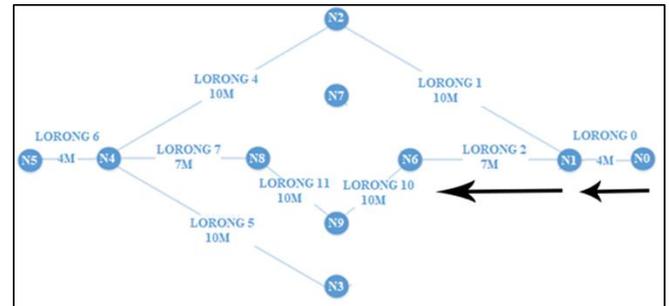
Tabel 4. Hasil Pengujian Kecocokan Data Terkirim dan Diterima

Pengujian ke	Data Terkirim	Output Lcd 16x2	Kesamaan Data
1	000100001100	000100001100	Sama
2	100011000010	100011000010	Sama
3	010000110101	010000110101	Sama
4	000000001111	000000001111	Sama
5	100000100000	100000100000	Sama
6	000011101100	000011101100	Sama
7	000000000000	000000000000	Sama
8	001000101000	001000101000	Sama
9	001000010000	001000010000	Sama
10	110001000101	110001000101	Sama

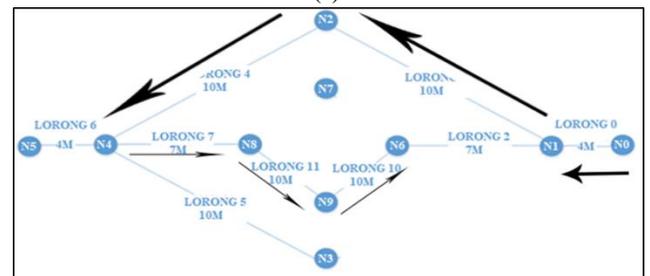
Pengujian Algoritma Dijkstra Dengan Antrian Prioritas Pada Aplikasi Tertanam

Pengujian ini dilakukan untuk mengetahui apakah algoritma dijkstra dengan antrian prioritas yang digunakan dapat mencari rute terpendek menuju jalur keluar. Data kerusakan yang digunakan pada pengujian ini adalah data yang ada pada Tabel 3. Pengujian dilakukan dengan memasukkan data kerusakan pada program, dan melihat panjang jarak rute keluar tiap simpul pada serial monitor Arduino IDE. Data panjang rute yang didapatkan kemudian dibandingkan dengan hasil perhitungan rute terpendek secara manual. Dari hasil perbandingan tersebut didapatkan apakah data jarak yang dihitung dengan aplikasi tertanam merupakan jarak terpendek atau tidak.

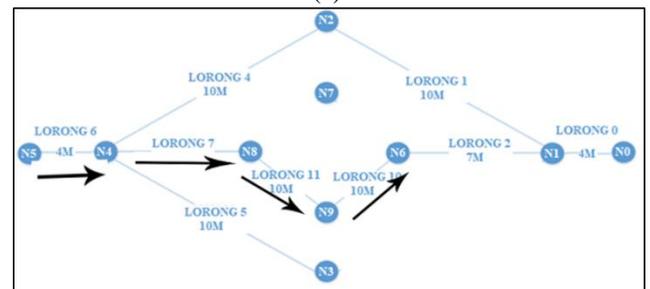
Jarak manual dihitung dengan cara menentukan simpul yang akan dituju kemudian menghitung setiap rute menuju simpul tujuan. Setelah itu, data rute akan dibandingkan dan diambil nilai terkecil. Contoh perhitungan manual menggunakan data kerusakan 1 dengan simpul 6 sebagai simpul tujuan. Ada 3 jalur yang dapat digunakan untuk mencapai simpul 6. Jalur pertama melalui simpul 0 – simpul 1 – simpul 6. Jalur pertama memiliki total jarak 11. Jalur kedua melalui simpul 0 – simpul 1 – simpul 2 – simpul 4 – simpul 8 – simpul 9 – simpul 6. Jalur kedua memiliki total jarak 51. Jalur ketiga melalui simpul 5 – simpul 4 – simpul 8 – simpul 9 – simpul 6. Jalur ketiga memiliki total jarak 24. Jalur terpendek yang didapatkan secara manual adalah jalur pertama dengan jarak 11. Proses ini dapat dilihat pada Gambar 10.



(a)



(b)



(c)

Gambar 10. (a) Jalur pertama; (b) Jalur kedua; (c) Jalur ketiga

Tabel 5. Hasil Pengujian Algoritma Dijkstra untuk Skenario Kerusakan 1

Skenario Kerusakan	Simpul	Jarak (m)	Jarak Manual (m)	Rute Terpendek
1	0	0	0	Benar
	1	4	4	Benar
	2	14	14	Benar
	3	14	14	Benar
	4	4	4	Benar
	5	0	0	Benar
	6	11	11	Benar
	7	-	-	Benar
	8	11	11	Benar
9	21	21	Benar	

Tabel 6. Hasil Pengujian Algoritma Dijkstra untuk Data Kerusakan 4

Skenario Kerusakan	Node	Jarak (m)	Jarak Manual (m)	Rute Terpendek
4	0	0	0	Benar
	1	4	4	Benar
	2	14	14	Benar
	3	14	14	Benar
	4	4	4	Benar
	5	0	0	Benar
	6	11	11	Benar
	7	-	-	Benar
	8	11	11	Benar
9	-	-	Benar	

Pada Tabel 5 dan Tabel 6 diperlihatkan hasil dari penghitungan jarak rute terpendek menuju semua simpul graf dengan algoritma dijkstra dengan antrian prioritas. Data tersebut kemudian dibandingkan dengan data jarak yang dihitung secara manual. Berdasarkan data pada Tabel 5 dan Tabel 6 didapatkan tingkat keberhasilan algoritma dijkstra dengan antrian prioritas untuk menghitung rute terpendek adalah 100%.

Pengujian Aplikasi Tertanam Untuk Menghidupkan LED

Pengujian ini dilakukan untuk memastikan bahwa program pada mikrokontroler dapat menghidupkan led sesuai dengan rute terpendek yang dihitung sebelumnya. Pengujian ini dilakukan menggunakan skenario kerusakan ke 4 pada Tabel 4. Setelah data kerusakan dimasukkan, led pada prototipe akan diperiksa apakah hidup dan mengarah ke pintu keluar yang terdekat. Hasil pengujian ini dapat dilihat pada Tabel 7 dan Gambar 11. Pada Gambar 11 ditampilkan data node atau simpul yang dilalui untuk menuju simpul berikutnya yang ditampilkan pada serial monitor Arduino ide. Berdasarkan hasil pengujian, pada gambar 12 node atau simpul yang tidak memiliki rute keluar akan menunjuk ke node 999. Artinya node tersebut tidak terhubung dengan pintu keluar karena kerusakan lorong.

Tabel 7. Hasil Pengujian Menghidupkan LED

Skenario Kerusakan	Led		Kondisi	Status
	Dari	Menuju		
4	Node 1	Node 0	Hidup	Benar
	Node 1	Node 2	Mati	Benar
	Node 1	Node 3	Mati	Benar
	Node 1	Node 6	Mati	Benar
	Node 2	Node 1	Hidup	Benar
	Node 2	Node 4	Hidup	Benar
	Node 3	Node 1	Hidup	Benar
	Node 3	Node 4	Hidup	Benar
	Node 4	Node 2	Mati	Benar
	Node 4	Node 3	Mati	Benar
	Node 4	Node 5	Hidup	Benar
	Node 4	Node 8	Mati	Benar
	Node 6	Node 1	Hidup	Benar
	Node 6	Node 7	Mati	Benar
	Node 6	Node 9	Mati	Benar
	Node 7	Node 6	Mati	Benar
Node 7	Node 8	Mati	Benar	
Node 8	Node 4	Hidup	Benar	
Node 8	Node 7	Mati	Benar	
Node 8	Node 9	Mati	Benar	
Node 9	Node 6	Mati	Benar	
Node 9	Node 8	Mati	Benar	



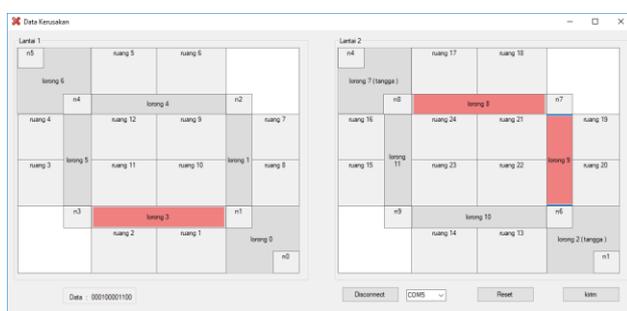
Gambar 11. Tampilan Serial Monitor Pengujian Menghidupkan LED Menggunakan Skenario Kerusakan ke 4

Data pada serial monitor menunjukkan rute keluar terpendek untuk tiap node. Selain itu terdapat lorong yang kondisinya tidak rusak tapi tidak dilewati karena bukan jalur terpendek. Hal ini dapat menjadi masalah apabila pada lorong tersebut ada korban yang ingin mengevakuasi diri keluar gedung. Hal ini terjadi karena algoritma dijkstra dengan antrian prioritas akan memilih satu jalur terpendek untuk tiap node dan mengabaikan jalur yang panjang. Untuk mengatasi ini, pada bagian program menghitung rute dimodifikasi agar menghitung beberapa jalur keluar. Pada bagian proses menghidupkan led juga ditambahkan baris program

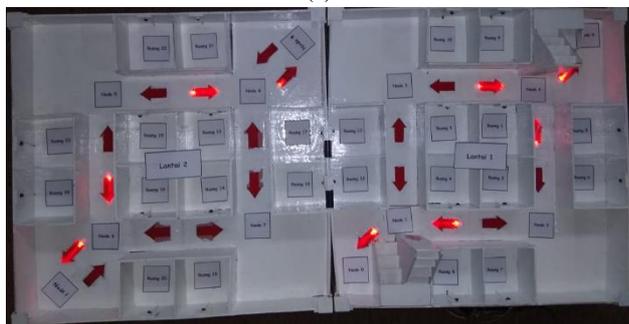
agar led pada lorong rute yang panjang tersebut juga hidup dan menunjuk ke arah rute keluar terdekat. Hal ini juga dapat membantu memecah jalur evakuasi sehingga proses evakuasi tidak macet dan dapat berjalan dengan lebih efisien. Dapat dilihat pada data di tabel 8 bahwa led hidup sesuai dengan data yang ditampilkan pada serial monitor. Selain itu, untuk setiap lorong yang kondisinya masih baik dan masih terhubung dengan pintu keluar, lednya hidup dan menunjuk ke arah pintu keluar.

Pengujian dan Analisa Sistem Secara Keseluruhan

Pengujian dan analisa sistem secara keseluruhan ini merupakan proses untuk memeriksa apakah sistem dapat berjalan dengan lancar dan benar sesuai yang diharapkan dari awal data kerusakan masuk sampai menampilkan jalur evakuasi. Pengujian ini dilakukan dengan cara memasukkan data kerusakan yang ada pada tabel 3 kemudian memeriksa apakah led pada prototipe menampilkan rute yang benar. Hasil pengujian ini dapat dilihat pada Gambar 12.



(a)



(b)

Gambar 12. Hasil Pengujian Sistem Menggunakan Skenario Kerusakan 1 (a) Data Kerusakan Pada Aplikasi Desktop; (b) Tampilan Led Pada Prototipe.

Pada Gambar 12 poin a menampilkan aplikasi desktop dengan skenario kerusakan 1 yaitu lorong 3, lorong 8 dan lorong 9 dalam keadaan rusak. Gambar 13 poin b menampilkan prototipe yang menunjukkan rute jalur keluar. Pada skenario ini, simpul 7 tidak terhubung dengan pintu keluar karena lorong penghubungnya rusak. Dari hasil pengujian sistem secara keseluruhan dapat dilihat bahwa led hidup sesuai data kerusakan yang diberikan. Dari 10 pengujian, semua led hidup dan menunjuk ke pintu keluar terdekat, kecuali untuk lorong yang rusak atau simpul yang tidak memiliki rute jalur keluar. Dari 10 kali pengujian sistem diatas didapatkan tingkat keberhasilan sistem dalam menunjukkan rute keluar terdekat sebesar 100%.

Pada Tabel 8 ditampilkan data waktu yang dibutuhkan bagi sistem untuk bekerja dari awal data kerusakan dimasukkan sampai led hidup. Data kerusakan yang digunakan adalah skenario kerusakan pada Tabel 2.

Tabel 9 Data Waktu Sistem Saat Dijalankan

Skenario Pengujian	Waktu
1	1097 ms
2	1096 ms
3	1096 ms
4	1098 ms
5	1098 ms
6	1097 ms
7	1100 ms
8	1098 ms
9	1099 ms
10	1097 ms
Rata-rata	1.6 ms

Dari tabel 8 didapatkan data waktu tercepat adalah 1096 ms sedangkan waktu terlama adalah 1100 ms. Sedangkan rata-rata waktu yang dibutuhkan oleh sistem saat mendapatkan data kerusakan sampai menghidupkan led adalah 1097.6 ms atau ± 1 detik. Waktu ini juga dipengaruhi oleh proses pengiriman data ke database yang dilakukan sebelum proses menghitung rute dilaksanakan. Data yang dikirim ke database digunakan untuk proses menyelamatkan korban yang tertinggal. Standar deviasi waktu proses sistem adalah 1.6 ms.

KESIMPULAN

Pada penelitian ini telah berhasil dibangun sistem yang dapat menentukan jalur evakuasi yang aman dan terpendek untuk keluar dari dalam gedung. Algoritma dijkstra dengan antrian prioritas akan mencari satu jalur terpendek untuk tiap node atau simpul dalam waktu rata-rata 3.6 ms untuk menjalankan algoritma dijkstra dengan antrian prioritas dan 0.2 ms untuk menghidupkan led. Waktu yang dibutuhkan oleh sistem untuk dapat berjalan secara keseluruhan dari awal sampai akhir adalah ± 1 detik. Didalam waktu 1 detik tersebut juga termasuk proses pengiriman data ke database menggunakan modul esp8266.

REFERENCES

- [1] Undang-undang nomor 24 tahun 2007 Tentang Penanggulangan Bencana
- [2] Abrahams, John. 1994. Chapter 6: Fire escape in difficult circumstances. In Stollard, Paul; Johnston, Lawrence. Design against fire: Introduction to Fire Safety Engineering Design. Chapman & Hall
- [3] Keputusan Menteri Kesehatan RI. No.228/Menkes/SK/1/1995 tentang Pembentukan Tim Kesehatan Penanggulangan Korban Bencana di setiap rumah sakit
- [4] Keputusan Menteri Kesehatan. KEPMENAKES 1087 MENKES / SK / VIII / 2010. Tentang Standar Kesehatan dan Keselamatan Kerja di Rumah Sakit.

- [5] Kementerian Kesehatan RI. 2012. Pedoman Teknis Bangunan Rumah Sakit Yang Aman Dalam Situasi Darurat dan Bencana.
- [6] Iftadi Irwan, dkk. Perancangan Peta Evakuasi Menggunakan Algoritma Floyd-Warshall untuk penentuan lintasan terpendek. Universitas sebelas maret: Surakarta. Jurnal Performa Vol. 10 No.2, 2011, hal. 95-104.
- [7] Munir, Rinaldi. 2010. Matematika Diskrit. Bandung:Informatika
- [8] Dewi, Luh Joni Erawati. 2010. Pencarian Rute Terpendek Tempat Wisata di Bali Dengan Menggunakan Algoritma Dijkstra. Universitas Pendidikan Ganesha:Yogyakarta.
- [9] Zhan, F. Benjamin. 1997. Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures. Universitas Southwest Texas State:San Marcos
- [10] T. Husna, D. Ichwana, dan Kasoep. Werman, "SISTEM PENGATUR IRIGASI SAWAH MENGGUNAKAN METODE," *Journal of Information Technology and Computer Engineering IRIGASI ALTERNATE WETTING AND DRYING BERBASIS TEKNOLOGI INTERNET OF THINGS*, vol. 02, pp. 42-50, 2018.
- [11] O. Octaviany, B. Rahmadya dan R. E. Putri, "Sistem Peringatan Dini Bencana Longsor Menggunakan Sensor Accelerometer dan Sensor Kelembabapan Tanah Berbasis Android," *Journal of Information Technology and Computer Engineering*, vol. 02, pp. 14-20, 2018.