PENCARIAN RUTE GEDUNG MENGGUNAKAN ALGORITMA DIJKSTRA

Lilis Kurniasari[1)], Mayadi[2)], Kusrini[3)]

[1) 2) 3)] Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta
Jl. Ring Road Utar, Condong Catur, Depok, Sleman, Yogyakarta 55281
Email : rainforest02@gmail.com[1)], mayadi.yadot2@gmail.com[2)], kusrini@amikom.ac.id[3)]

Abstrak

Navigasi merupakan kebutuhan mendasar setiap individu. Setiap orang memiliki kriteria sendiri dalam mendefinisikan dengan baik dalam memilih jalur atau rute ketika bergerak dari satu tempat ke tempat yang lain. Paper ini menyajikan simulasi pencarian rute terpendek menggunakan algoritma dijkstra, studi kasus pada komplek kampus terpadu Universitas Muhammadiyah Yogyakarta. Paper ini dirancang untuk membuat jalur atau rute terpendek dalam pencarian gedung pada komplek kampus terpadu Universitas Muhammadiyah Yogyakarta.

Kata kunci: *Shortestpath, Algoritma Dijkstra, Matlab Simulink.*

1. Pendahuluan

Navigasi merupakan kebutuhan mendasar setiap individu, setiap orang memiliki kriteria tersendiri dalam mendefinisikan dengan baik untuk memilih jalur atau jalan ketika bergerak dari satu tempat ke tempat yang lain [1]. Untuk orang orang yang sering mengalami masalah dengan pencarian tempat, penentuan jalur terpedek dapat memegang peranan yang sangat penting. Untuk saat, kebanyakan metode perencanaan jalan menggunakan jarak terpendek atau biaya paling sedikit sebagai standar utama [2]. Seperti pada kasus pencarian lokasi gedung ujian tertulis penerimaan siswa baru dikampus terpadu Universitas Muhammadiyah Yogyakarta (UMY). Sering kali peserta ujian tertulis mengalami kesulitan dalam menemukan gedung lokasi tempat ujian tertulis diadakan. Navigasi yang baik akan memberikan kenyamanan pada peserta ujian sehingga peserta ujian dapat dengan mudah menemukan lokasi tempat ujian berada [1]. Penelitian ini bertujuan untuk membuat navigasi untuk menentukan rute atau jalur terpendek menggunakan algoritma dijkstra studi kasus pada komplek gedung kampus terpadu UMY.

Ada beberapa algoritma yang digunakan dalam pencarian jalur atau rute terpendek, algortima tersebut terbagi menjadi dua metode yaitu tradisional dan heuristic [3]. Menurut beberapa penelitian analisis dan perbandingan algoritma djisktra menjadi algoritma lebih baik dalam penentuan jalur terpendek [4]. Kompleksitas algoritma Dijkstra adalah dapat diterima dalam beberapa kasus jarak jalur pendek, jadi makalah ini memilih Algoritma Dijkstra sebagai algoritma dasar perencanaan jalur terpendek. Algoritma Dijkstra diakui sebagai algoritma klasik yang digunakan untuk menimbang jalur terpendek, dan juga yang paling sempurna teori, algoritma yang paling banyak digunakan, yang menghitung jalur terpendek dari jaringan berat nonnegatif [5].

Ada beberapa penelitian yang menggunakan algoritma dijkstra dan aplikasi praktis dalam memecahkan masalah perencaan rute terpendek, salah satunya penelitian untuk memecahkan permasalahan jalan dilingkungan persegi panjang untuk mendapatkan jalur optimal dengan jarak dan waktu terpendek dengan menambahkan waktu berjalan evaluasi perencanaan jalan [6]. Jehyun Cho dalam penelitiannya mengembangkan smart system untuk mendeteksi daerah berbahaya secara real time and memberikan jalur terpendek dan aman menuju tempat evakuasi menggunakan algoritma dijkstra [7]. Tahun 2014 peneliti dari universitas Tsinghua Beijing menggunakan algoritma dijkstra untuk membangun model evakuasi dalam keadaan darurat menggunakan kendaran [8]. Dalam penelitian tersebut diusulkan jalur evakuasi optimal menggunakan tiga kasus yang berbeda. Hasil yang diperoleh memberikan metode prediksi dan teoritis yang baik untuk pemilihan jalur evakuasi darurat terutama bagi mereka yang tinggal dilingkungan dengan tingkat kepadata yang tinggi. Aram M. Ahmed dalam makalahnya menggunakan algoritma dijkstra untuk memudahkan wisatawan dalam mencari hotel di daeraha wisata. Selain itu wisataraan juga dapat mengetahui rute tercepat ke semua tempat wisata dan restoran terdekat dengan hotel dimana mereka tinggal [9].

Algoritma dijkstra ditemukan oleh Edsger Dijkstra seorang ilmuan komputer dari Belanda pada tahuan 1959. Algoritma dijkstra digunakan untuk pencarian grafik untuk memecahkan masalah jalur terpendek menuju sumber tunggal untuk grafik dengan tepi tidak negatif (*nonnegative*). Sebagai contoh, jika simpul grafik mewakili kota dan biaya jalur tepi mewakili jarak mengemudi antara pasang kota yang terhubung dengan jalan langsung, algoritma Dijkstra bisa digunakan untuk menemukan rute terpendek antara satu kota dan kota lainnya.

Metodologi yang digunakan untuk menemukan jalur terbaik dengan menggunakan algoritma Dijkstra adalah sebagai berikut:

1. Tetapkan ke setiap node dengan nilai jarak dan tetapkan nol ke simpul awal dan ke tak terhingga untuk semua node lainnya.
2. Tandai semua node sebagai simpul awal yang belum dikunjungi dan setel sebagai arus.

3. Untuk node saat ini, pertimbangkan semua tetangga yang belum dikunjungi dan hitung nilainya jarak tentatif (dari simpul awal). Misalnya, jika current node (A) mempunyai jarak 6, dan edge yang menghubungkannya dengan node lain (B) adalah 2, yaitu jarak ke B melalui A akan menjadi 6 + 2 = 8. Jika jarak ini kurang dari sebelumnya tercatat jarak (tak terhingga di awal, nol untuk awal node), menimpa jarak.
4. Saat mempertimbangkan semua tetangga simpul saat ini, tandai seperti yang dikunjungi. Sebuah node yang dikunjungi tidak akan diperiksa lagi; jarak yang direkam sekarang bersifat final dan minimal.
5. Jika semua node telah dikunjungi, selesai. Jika tidak, setel node yang belum dikunjungi jarak terkecil (dari simpul awal) sebagai "node arus" berikutnya dan lanjutkan dari langkah 3.

Dijkstra merupakan bagian dari algoritma Greedy, yang biasa digunakan untuk mencari nilai maksimal atau nilai minimum [7]. Algoritma ini akan menghitung iterasi jarak antara titik awal ke titik lainnya dalam jaringan secara bergantian, dan bergantung pada jalan mana yang akan digunakan untuk melakukan perjalanan dari titik awal ke titik yang lain. jarak akan bervariasi. Titik awalnya memiliki nilai jarak 0. Algoritma menggantikan jarak nilai disimpan pada setiap titik (node) dari titik awal ke setiap titik dengan nilai baru jika jalan yang lebih pendek ditemukan .Algoritma Dijkstra telah digunakan di berbagai bidang, seperti Google Maps, sistem navigasi untuk mobil, teknik lalu lintas, robotika, dan teknik Industri. Seperti metode pencarian yang lain algoritma dijkstra juga menggunakan graph untuk menentukan keadaan awal pencaraian.

Graf merupakan pokok bahasan yang sudah tua usianya namun memiliki banyak terapan dalam kehidupan sehari-hari sampai saat ini. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Banyak persoalan pada dunia nyata yang sebenarnya merupakan representasi visual dari graf. Contoh salah satu representasi visual dari graf adalah peta.

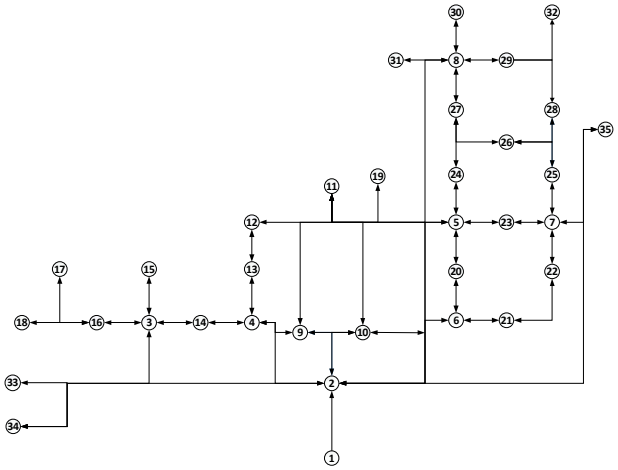
2. Pembahasan

Penelitian ini mengembangkan sebuah system untuk pencarian rute terdekat, system ini dapat memberikan navigasi untuk mencapai lokasi atau gedung tempat ujian tertulis dilaksanakan dengan menggunakan algoritma dijkstra. Pengguna akan dipandu oleh system untuk menemukan rute menuju gedung tujuan.

Rancangan awal penelitian ini dimulai dari pengumpulan data yang dibutuhkan untuk peneltian yang pertama adalah data denah lokasi gedung di kampus terpadu umy seperti terlihat pada gambar 2 dan 3.

Gambar 1. *Map kampus terpadu UMY*

Tanda panah bolak balik menandakan bahwa jalur tersebut dapat dilalui dari dua arah, sebagai contoh tanda panah dari point 1 ke point 2 berarti jalur tersebut dapat dilalui dari point dari point satu menuju point dua dan sebaliknya dari point 2 menuju point 1. Dari gambar 2 dapat dilihat bahwa semua jalur dapat dilalui dari dua arah.



Gambar 2. *Denah lokasi komplek gedung UMY*

Dari gambar 2 dapat dijabarkan bahwa setiap point atau node merepresestasikan gedung yang ada di komplek UMY, detail mengenai representasi gedung dapat dilihat pada tabel 1.

Tabel 1. *Representasi node*

Point	Lokasi
1	Gerbang Utama
2	Percabangan
3	Hall komplek Gd E
4	Simpang Gd E2 dan E4
5	Hall komplek Gd F
6	Simpang Gd F1 dan F4
7	Simpang Gd F3, F5, dan F6
8	Hall Komplek Gedung G
9	Rektorat A
10	Rektorat B
11	Masjid Ahmad Dahlan
12	Gedung D Perpustakaan
13	Gedung E4
14	Gedung E2
15	Gedung E3
16	Gedung E1
17	Gedung E5
18	Gedung E6
19	Gedung Pasca Sarjana

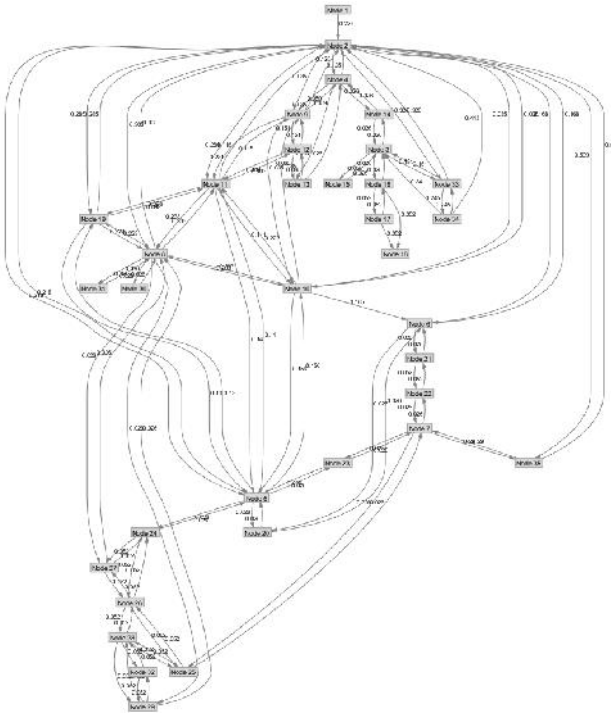
Point	Lokasi
20	Gedung F1
21	Gedung F4
22	Gedung F5
23	Gedung F3
24	Gedung F2
25	Gedung F6
26	Gedung F7
27	Gedung G1
28	Gedung G2
29	Gedung G3
30	Gedung G4
31	Gedung G5
32	Gedung G6
33	Sportorium
34	Gerbang Selatan
35	Gerbang Utara

Langkah kedua adalah menentukan jarak antar gedung atau jarak antar node seperti terlihat pada tabel 2.

Tabel 2. *Jarak antar gedung (titik/poin)*

Point		Jarak	Point		Jarak	Point		Jarak	Point		Jarak
1	2	22.7	7	22	2.6	12	9	15.1	25	28	5.2
2	9	3.5	7	23	2.6	12	11	10.3	26	24	5.2
2	10	3.5	7	25	2.6	12	13	6.5	26	25	5.2
2	11	28.1	8	2	30.5	13	4	2.6	26	27	5.2
2	4	12.5	8	10	20.3	13	12	6.5	26	28	5.2
2	5	21.8	8	11	23,3	14	4	2.6	27	8	2.6
2	6	16.8	8	19	22.3	14	3	2.6	27	24	5.2
2	8	30.5	8	27	2.6	15	3	2.6	27	26	5.2
2	33	32.6	8	29	2.6	16	18	5.2	28	25	5.2
2	35	59.9	8	30	2.6	16	17	5.2	28	26	5.2
2	19	28.5	8	31	2.6	16	3	2.6	28	29	5.2
3	14	2.6	9	2	3.5	17	16	5.2	28	32	5.2
3	15	2.6	9	4	5.9	17	18	5.2	29	8	2.6
3	16	2.6	9	10	3.5	19	2	28.5	29	28	5.2
3	33	15.1	9	11	11.5	19	5	13	29	32	5.2
4	2	12.5	9	12	15.1	19	8	22.3	30	8	2.6
4	9	5.9	10	2	3.5	19	11	9.8	31	8	2.6
4	13	2.6	10	5	15.6	20	6	2.6	32	28	5.2
4	14	2.6	10	6	10.5	20	5	2.6	32	29	5.2
5	2	21.8	10	8	20.3	21	6	2.6	33	2	32.6
5	10	15.6	10	9	3.5	21	22	5.2	33	3	15.1
5	11	14	10	11	11.1	22	7	2.6	33	34	24.5
5	19	13	10	11	11.1	22	21	5.2	34	2	44.3
5	20	2.6	11	2	28.1	23	7	2.6	34	3	34
5	23	2.6	11	12	10.3	23	5	2.6	34	33	24.5
5	24	2.6	11	19	9.8	24	5	2.6	35	2	59.9
6	2	16.8	11	5	14	24	27	5.2	35	7	29
6	20	2.6	11	8	23,3	24	26	5.2			
6	21	2.6	11	9	11.5	25	7	2.6			
7	35	29	11	10	11.1	25	26	5.2			

Dari data yang terkumpul langkah selanjutnya adalah pembuatan graph dari semua node yang terhubung dan mensimulasikannya menggunakan aplikasi Matlab seperti terlihat pada gambar 4.



Gambar 4. *Graph lokasi gedung UMY*

Untuk memverifikasi kelayakan sistem, pengujian dalam penelitian dilakukan dengan 3 kali simulasi pencarian. Simulasi pertama dari pintu masuk menuju komplek gedung. Simulasi kedua komplek gedung menuju pintu keluar. Simulasi ketiga pencarian antar gedung. Parameter yang dgunakan dalan pengujian ada 2 yaitu jarak (*Distance*) dan rute (*Path*). Untuk mendapatkan data yang lebih akurat, dalam penelitian ini juga menggunakan metode pencarian yang lain dalam simulasi pencarian rute. Dalam penelitian ini metode yang digunakan untuk komparasi adalah Breath First Search (BFS).

node awal dan node yang akan digunakan dalam simulasi sebagai berikut:

- Simulasi pertama
  Dimulai dari node 1 sebagai node awal dan node 11 sebagai node tujuan.
- Simulasi kedua
  Node 30 sebagai node awal dan node 35 sebagai node tujuan.
- Simulasi ketiga
  Node 14 sebagai node awal dan node 32 sebagai node akhir.

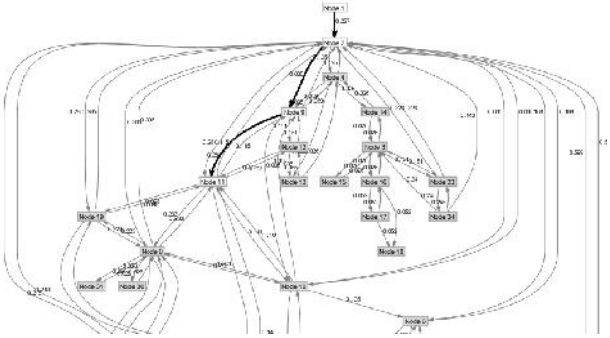
Hasil simulasi pencaraian dapat dilihat secara detail pada tabel 3 dan path diagram dapat dilihat pada gambar 5, 6, 7 dan 8.

Tabel 3. *Hasil simulasi pencarian rute*

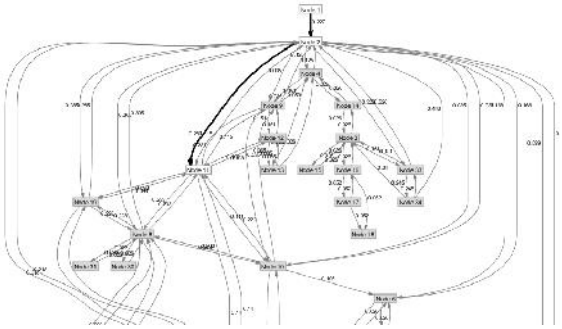
Simulasi	Node Awal	Node Tujuan	Dijkstra		BFS	
			Jarak	Rute	Jarak	Rute
Pertama	1	11	0.377	1 2 9 11	2	1 2 11
Kedua	30	35	0.472	30 8 27 26 25 7 35	3	30 8 2 35
Ketiga	14	32	0.401	14 4 9 10 8 29 32	5	14 4 2 8 29 32

Tampak dari tabel untuk 3 simulasi yang dilakukan dari node yang berbeda bahwa perbandingan hasil pencarian rute menggunakan algoritma dijkstra dan BFS dapat disimpulkan sebagai berikut:

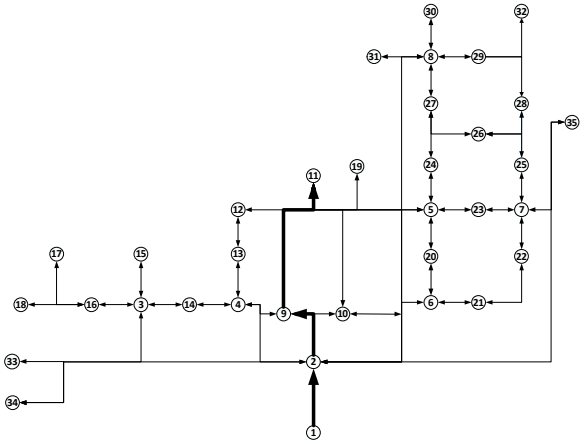
1. Secara umum simulasi menggunakan algoritma dijkstra mempunyai path atau rute yang dilewati lebih banyak dari pada metode BFS, hal ini disebabkan pada metode pencarian dijkstra akan menandai semua node sebagai simpul awal yang belum dikunjungi. Kemudian akan menyimpan node dengan jarak paling pendek, begitu seterusnya sampai ditemukan node tujuan. Sedangkan metode BFS akan mengunjungi suatu node kemudian mengunjungi semua node yang bertetangga dengan tersebut tersebut terlebih dahulu begitu seterusnya sehingga didapatkan node tujuan. Dari gambar dibawah dapat dijelaskan bahwa node 2 bertengga dengan node 11, sehingga didapatkan rute yang lebih sedikit.
2. Jarak yang dihasilkan dari ketiga simulasi diatas terlihat bahwa simulasi dengan algoritma djikstra mempunyai jarak yang lebih pendek dari pada metode BFS, hal ini disebabkan saat mengunjungi node node, algoritma djikstra hanya menyimpan node yang mempunyai jarak minimal. Setalah node tujuan ditemukan algoritma dijkstra akan mengkalkulasi semua jarak minimal antar node yang sudah disimpan. Sedangkan metode BFS mempunyai jarak yang lebih jauh hal ini disebabkan metode BFS tidak melihat jarak yang dtempuh namun metode ini hanya mencari node tujuan dengan cara mengunjungi tetangga terdekat lebih dahulu.



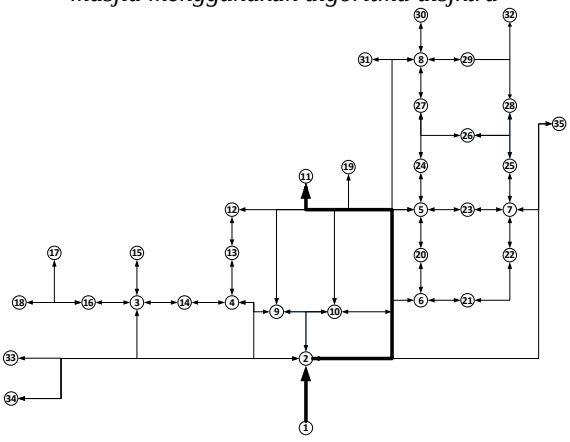
Gambar 5. *Rute node 1 node 11 dijkstra*



Gambar 6. *Rute node 1 node 11 BFS*



Gambar 7. *Rute dari pintu gerbang utama menuju masjid menggunakan algoritma disjktra*



Gambar 8. *Rute dari pintu gerbang utama menuju masjid menggunakan algoritma BFS*

Seperti disebutkan dalam tujuan penelitian yaitu untuk membuat navigasi untuk menentukan rute atau jalur terpendek menggunakan algoritma dijkstra studi kasus pada komplek gedung kampus terpadu UMY. Dari simulasi yang telah dilakukan memperlihatkan tujuan penelitian tercapai. Gedung tujuan dapat ditemukan melalui jalur terpendek menggunakan algoritma dijkstra, sedangkan dengan metode BFS didapatkan jarak yang lebih panjang.

3. Kesimpulan

Setelah diamati dari analisis peneltian diketahui bahwa untuk memperoleh jarak terpendek menemukan gedung sekali lagi membuktikan algoritma dijkstra lebih unggul. Disisi yang lain Google map juga menggunakan algoritma dijkstra. Google juga menyediakan API dari Google map untuk dapat diintegrasikan kedalam aplikasi yang berbasis navigasi atau pencarian rute, sehingga system ini dapat diwujudkan kedalam bentuk aplikasi dengan memanfaatkan teknologi API dari Google map. Dengan algoritma dijkstra kita dapat membuat navigasi untuk membatu peserta ujian menemukan gedung lokasi tempat ujian berada dengan sangat mudah. Kekurangan penelitian ini belum ada parameter untuk perhitungan waktu dan hanaya menggunakan satu algoritma lan sebagai komparasi Penelitian selanjutnya lebih mengarah pada pencarian rute terpendek untuk mencari ruangan tempat ujian tertulis dilaksanakan.

Daftar Pustaka

[1] K. S. Durgesh Nandini, "A novel path planning algorithm for visually impaired people," *Journal of King Saud University – Computer and Information Sciences*, vol. xxx, 2017.

[2] L. Rosyidi, "Timebase dynamic weight for Dijkstra Algorithm implementation in route planning software," *International Conference on Intelligent Green Building*, 2014.

[3] G. X. Yujin, "Optimal Route Planning of Parking Lot Based on Dijkstra Algorithm," in *International Conference on Robots & Intelligent System*, 2017.

[4] Y. Xu, "Indoor optimal path planning based on Indoor optimal path planning based onIndoor optimal path planning based on," in *International Conference on Materials Engineering and Information Technology Applications*, 2015.

[5] Q. Guo, "Path-planning of automated guided vehicle based on improved Dijkstra algorithm," in *guided vehicle based on improved Dijkstra algorithm*, 2017.

[6] J. Choa, "Application of Dijkstra's Algorithm in the Smart Exit Sign," *The 31st International Symposium on Automation and Robotics in Construction and Mining*, 2014.

[7] C. Yi-zhou, "Path Optimization Study for Vehicles Evacuation Based on Dijkstra algorithm," *Procedia Engineering*, p. 159 – 165, 2014.

[8] J. L, "Optimization Studies Based on Shortest Path Algorithm of Optimization Studies Based on Shortest Path Algorithm of," *Journal of Weinan Teachers University*, 2009.

[9] A. M. Ahmed, "Dijkstra Algorithm Applied: Design and Implementation of a framework to find nearest Hotels and Booking Systems in Iraqi," in *International Conference on Current Research in Computer Science and Information Technology*, Iraq, 2017.

Biodata Penulis

Lilis Kurniasari, memperoleh gelar Sarjana Teknik (S.T), Jurusan Teknik Elektro Universitas Muhammadiyah Yogyakarta, lulus tahun 2008. sedang menempuh pendidikan di Magister Tenik Informatika Universitas AMIKOM Yogyakarta.

Mayadi , memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika di STMIK BUMIGORA MATARAM, lulus tahun 2016. Saat ini sedang menempuh pendidikan di Magister Tenik Informatika Universitas AMIKOM Yogyakarta.

Kusrini , memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Ilmu Komputer Universitas Gadjah Mada, lulus tahun 2002. Memperoleh gelar Master Komputer (M.Kom), Jurusan Ilmu Komputer Universitas Gadjah Mada, lulus tahun 2006. Memperoleh gelar Doktor (Dr), Jurusan Ilmu Komputer Universitas Gadjah Mada, lulus tahun 2010. Saat ini menjadi Dosen dan Direktur Program Pascasarjana di Universitas AMIKOM Yogyakarta..

ISSN : 2302-3805