

Perbandingan Algoritme Dijkstra dan Algoritme A* (A-Star) dalam Penentuan Lintasan Terpendek dari Dinas Pendidikan Provinsi Lampung ke Beberapa Sekolah Menengah Atas (SMA) Negeri di Provinsi Lampung

¹Silvia Rizki Aulia, ^{*2}Wamiliana, ³Asmiati, dan ⁴Notiragayu

^{1,2,3,4} Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung
Jalan Soemantri Brojonegoro No.1 Gedung Meneng, Bandar Lampung, Provinsi Lampung, Indonesia
Email: ¹silviarizkiaulia@gmail.com, ^{*2}wamiliana.1963@fmipa.unila.ac.id, ³asmiati.1976@fmipa.unila.ac.id,
⁴notiragayu@fmipa.unila.ac.id

Abstract — *In this research, we aim to find the most efficient route from the Lampung Provincial Education Office to various State Senior High Schools in Lampung Province, employing both the Dijkstra Algorithm and the A* (A-Star) Algorithm. To address this problem, we model it as a weighted graph, where the vertices represent the locations of the Lampung Provincial Education Office and the Public High Schools, and the edges' weights represent the distances. Furthermore, we implement the solution using the Python programming language. The outcomes demonstrate that both algorithms yield optimal results with the same distance and route. However, when considering the execution time of the program, the A* (A-Star) algorithm outperforms Dijkstra's Algorithm because it quickly identifies the shortest path from the starting point to the destination, whereas Dijkstra's Algorithm explores the shortest paths from neighboring points first.*

Keywords: A*(A-Star) Algorithm; Dijkstra Algorithm; Shortest Path.

1. PENDAHULUAN

Dalam era teknologi saat ini, pencarian lokasi yang diinginkan telah menjadi lebih mudah berkat perkembangan teknologi, terutama dengan layanan seperti Google Maps. Pengguna dapat dengan mudah menemukan lokasi, menghitung jarak, dan memperkirakan waktu tempuh sesuai dengan preferensi mereka. Namun, dalam pencarian lintasan terpendek, terutama untuk tujuan pengiriman atau perjalanan, efisiensi sangat penting, baik dari segi waktu maupun jarak yang harus ditempuh. Untuk mencapai tujuan ini, algoritme seperti Dijkstra dan A* (A-Star) digunakan dalam penelitian sebelumnya. Hasil penelitian tersebut menunjukkan bahwa algoritme A* menghasilkan nilai yang lebih kecil dibandingkan dengan perhitungan manual, menunjukkan efektivitasnya dalam menentukan lintasan terpendek [1].

Provinsi Lampung, yang terdiri dari 15 kabupaten/kota dan 229 kecamatan, memiliki banyak fasilitas pendidikan yang tersebar di seluruh wilayahnya. Data Pokok Pendidikan Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi tahun 2022 mencatat adanya 239 Sekolah Menengah Atas (SMA) Negeri di provinsi ini. Dalam konteks penelitian ini, ditentukan lintasan terpendek dari Dinas Pendidikan Provinsi Lampung ke beberapa Sekolah Menengah Atas (SMA) Negeri yang ada di berbagai kabupaten/kota di provinsi tersebut. Sebagai representasi, penelitian ini mengambil dua SMA Negeri yang memiliki lintasan terpendek dari Dinas Pendidikan Provinsi Lampung.

Penelitian ini menunjukkan bagaimana teknologi dan algoritme perhitungan lintasan terpendek seperti A* dapat sangat berguna dalam mengoptimalkan perencanaan perjalanan atau pengiriman, terutama dalam konteks pendidikan di Provinsi Lampung. Dengan memahami lintasan terpendek dari Dinas Pendidikan ke sekolah-sekolah di wilayah tersebut, diharapkan dapat membantu meningkatkan efisiensi dan pengelolaan sumber daya, yang pada gilirannya akan berdampak positif pada penyediaan pendidikan di daerah tersebut.

2. METODOLOGI PENELITIAN

2.1 Masalah Lintasan Terpendek

Suatu graf $G(V, E)$ didefinisikan sebagai pasangan terurut (V, E) dengan V adalah himpunan berhingga yang tak kosong dan memuat elemen-elemen yang disebut *vertex* atau titik, dan E adalah himpunan elemen-elemen (mungkin kosong) yang berbentuk garis atau disebut *edge* yang menghubungkan titik di G . Jalan (*walk*) adalah barisan berhingga yang berselang-seling dari titik dan garis, sedemikian sehingga setiap garis menempel dengan titik sebelum dan sesudahnya. Pada jalan untuk titik dan garis diperbolehkan pengulangan. Lintasan (*path*) adalah suatu jalan dimana tidak ada titik yang dilewati lebih dari satu kali dan jika titik awal dan akhir sama maka lintasan tersebut disebut lintasan tertutup [2]. Persoalan lintasan terpendek (*Shortest Path Problem*) merupakan satu persoalan optimasi yang menggunakan graf berbobot, dimana bobot pada setiap sisi graf tersebut dapat digunakan untuk menyatakan jarak antar kota, waktu pengiriman pesan, ongkos pembangunan, dan sebagainya.

2.2. Euclidean

Euclidean adalah fungsi heuristik yang diperoleh berdasarkan jarak langsung bebas hambatan. Sebelum mendapatkan nilai fungsi heuristik, kedua titik harus direpresentasikan ke dalam koordinat 2 dimensi (x, y) . Hasil perhitungan masih dalam satuan *decimal degree* (sesuai dengan format latitude-longitude yang dipakai) sehingga untuk menyesuaikannya perlu dikalikan dengan 111.319 km (1 derajat bumi = 111.319 km). Persamaan Euclidean dapat dilihat pada Persamaan 1. Perhitungan lintasan terpendek menggunakan algoritme A* dapat dilakukan setelah didapat nilai heuristik dari masing-masing titik [3].

$$H_{a-b} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

Keterangan:

- H_{a-b} : estimasi jarak a ke b
- x_1 : koordinat x titik b
- y_1 : koordinat y titik b
- x_2 : koordinat x titik a
- y_2 : koordinat y titik a

2.3 Pemrograman Python

Python diciptakan oleh Guido van Rossum di Belanda pada tahun 1990 dan namanya diambil dari acara televisi kesukaan *Guido Monty Python's Flying Circus*. Kemudian *Python* menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintaks intuitif, dan memiliki pustaka yang luas [4][5].

2.4. Algoritme Dijkstra

Algoritme Dijkstra merupakan algoritme yang digunakan untuk menentukan jarak terpendek dari satu titik ke titik yang lainnya pada suatu graf berbobot, jarak antar titik adalah nilai bobot dari setiap garis pada graf. Suatu bobot harus bernilai positif (bobot ≥ 0). Algoritme Dijkstra ditemukan oleh Edger Wybe Dijkstra. Cara kerja algoritme Dijkstra dalam pencarian jarak terpendek adalah perhitungan dari titik asal ke titik terdekatnya, kemudian ke titik yang kedua, dan seterusnya. Algoritme Dijkstra telah diimplementasikan antara lain oleh [6] untuk menentukan objek kota Medan, dan oleh [7] untuk menentukan rute terpendek ke museum di Jakarta. Secara umum algoritme Dijkstra ini dapat digambarkan sebagai berikut:

Misalkan N = himpunan titik-titik dimana lintasan terpendek telah didapat, s = titik awal
Inisiasi: $N = \{s\}$, $d_s = 0$ (jarak dari s ke dirinya sendiri adalah 0), $d_j = c_{sj}$ untuk setiap $j \neq s$
Langkah A: (Menentukan titik terdekat i).
Tentukan $i \notin N$ sehingga $d_i = \min d_j$ untuk $j \notin N$

Tambahkan i ke N .

Jika N memuat semua titik, stop.

Langkah B: (Meng-*update* jarak terkecil

Untuk setiap titik $j \notin N$

$d_j = \min(d_j, d_i + c_{ij})$

Kembali ke Langkah A

2.5. Algoritme A* (A-Star)

Algoritme A* adalah algoritme pencarian lintasan terpendek (*shortest path*) yang merupakan perbaikan dari Algoritme BFS (*Best First Search*) dengan memodifikasi fungsi heuristiknya untuk memberikan hasil yang optimal. Algoritme A* telah digunakan antara lain oleh [8] untuk menentukan tarif minimum taksi bandara berdasarkan rute terpendek, serta [9] dalam menentukan rute terpendek tempat industri di Provinsi Banten. Algoritme A* menginvestigasi setiap titik yang ada dalam observasi. Untuk setiap titik, jarak yang ada ditambahkan dengan fungsi $h(n)$ yang merupakan nilai heuristik untuk mencapai titik n . Selain itu digunakan daftar himpunan O (*open*) untuk titik-titik yang belum dihitung jaraknya, dan C (*close*) untuk titik yang telah dihitung. Nilai jarak ditentukan oleh $f(n) = g(n) + h(n)$.

Langkah-langkah algoritme A* [10] secara garis besar diberikan sebagai berikut:

1. Masukkan titik awal ke list *OPEN*.
2. Ulangi
 - a. Cari titik (n) dengan nilai $f(n)$ yang paling kecil dalam list *OPEN list* (titik menjadi *current node*).
 - b. Pindahkan *current node* ke list *CLOSE*
 - c. Untuk setiap tetangga dari *current node* lakukan berikut:
 - (i) Jika tidak dapat dilalui atau sudah ada dalam list *CLOSE*, abaikan.
 - (ii) Jika belum ada di list *OPEN* buat *current node parent* dari titik tetangga ini.
 - (iii) Jika sudah ada di list *OPEN* cek apabila titik tetangga ini lebih baik. Jika ya, ganti *parent* dari titik ini di list *OPEN* jadi *current node*, hitung ulang.
3. Hentikan pengulangan jika :
 - (i) List *OPEN* telah memuat titik tujuan.
 - (ii) List *OPEN* kosong (tidak ada solusi).

2.6. Data yang Digunakan

Data yang digunakan merupakan data yang diperoleh dari Google Maps untuk lokasi Dinas Pendidikan Propinsi Lampung dan 30 SMA Negeri yang ada di beberapa kota di Provinsi Lampung. Langkah-langkah yang dilakukan dalam penelitian ini sebagai berikut:

- a. Menentukan objek yang akan dicari penyelesaian lintasan terpendeknya.
- b. Memodelkan titik-titik dari objek yang akan direpresntasikan menjadi suatu graf.
- c. Melakukan perhitungan secara manual untuk menentukan lintasan terpendek.
- d. Mengumpulkan data berupa jarak antar titik lokasi pada graf yang untuk diinput ke *Python*.
- e. Mencari solusi dari masing-masing algoritme dengan input data yang sudah ada.
- f. Menganalisis dan menyimpulkan perbedaan hasil/*output* dari kedua algoritme, baik secara manual maupun dengan menggunakan bantuan pemogramam *Python*.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini digunakan 31 titik lokasi dengan tujuan 30 SMA Negeri di Lampung dan Dinas Pendidikan Provinsi Lampung sebagai titik awal. Lokasi yang diambil yaitu masing-masing 2 titik lokasi per kabupaten yang ada di Provinsi Lampung. Daftar titik lokasi yang diambil diilustrasikan pada Tabel 1.

Tabel 1. Daftar 31 lokasi dan titik-titik yang mewakilinya.

No.	Lokasi	Titik
1.	Dinas Pendidikan Provinsi Lampung	$A = v_1$
2.	SMAN 2 Bandar Lampung	$C = v_2$
3.	SMAN 12 Bandar Lampung	$Q = v_3$
4.	SMAN 1 Gedong Tataan	$S = v_4$
5.	SMAN 1 Way Khilau	$T = v_5$
6.	SMAN 1 Pringsewu	$U = v_6$
7.	SMAN 1 Pardasuka	$V = v_7$
8.	SMAN 1 Metro	$W = v_8$
9.	SMAN 5 Metro	$X = v_9$
10.	SMAN 1 Gunung Sugih	$Y = v_{10}$
11.	SMAN 1 Way Seputih	$Z = v_{11}$
12.	SMAN 1 Kalianda	$AA = v_{12}$
13.	SMAN 1 Bakauheni	$AB = v_{13}$
14.	SMAN 1 Sukadana	$AC = v_{14}$
15.	SMAN 1 Pasir Sakti	$AD = v_{15}$
16.	SMAN 1 Kotaagung	$AE = v_{16}$
17.	SMAN 1 Gunung Alip	$AF = v_{17}$
18.	SMAN 1 Blambangan Umpu	$AG = v_{18}$
19.	SMAN 1 Buay Bahuga	$AH = v_{19}$
20.	SMAN 1 Kota Bumi	$AI = v_{20}$
21.	SMAN 1 Bukit Kemuning	$AJ = v_{21}$
22.	SMAN 1 Tulang Bawang Tengah	$AK = v_{22}$
23.	SMAN 1 Gunung Agung	$AL = v_{23}$
24.	SMAN 1 Menggala	$AM = v_{24}$
25.	SMAN 1 Penawartama	$AN = v_{25}$
26.	SMAN 1 Mesuji	$AO = v_{26}$
27.	SMAN 1 Rawajitu Utara	$AP = v_{27}$
28.	SMAN 1 Bengkuntan	$AQ = v_{28}$
29.	SMAN 1 Pesisir Selatan	$AR = v_{29}$
30.	SMAN 1 Liwa	$AS = v_{30}$
31.	SMAN 1 Belalau	$AT = v_{31}$

Tabel 2 berikut adalah perhitungan nilai $h(n)$ berdasarkan titik koordinat masing-masing titik lokasi yang didapat menggunakan Google Maps menggunakan Persamaan 1 dengan tujuan $AB = v_{13}$.

Tabel 2. Perhitungan nilai $h(n)$ dengan tujuan titik AB.

No.	Titik Asal	Nilai $h(n)$ (km)
1.	$A = v_1$	69,597
2.	$C = v_2$	70,759
3.	$Q = v_3$	70,48
4.	$S = v_4$	86,711
5.	$T = v_5$	93,848
6.	$U = v_6$	101,069
7.	$V = v_7$	98,318
8.	$W = v_8$	94,015
9.	$X = v_9$	96,211
10.	$Y = v_{10}$	121,926
11.	$Z = v_{11}$	121,633
12.	$AA = v_{12}$	20,855
13.	$AB = v_{13}$	0
14.	$AC = v_{14}$	91,445
15.	$AD = v_{15}$	39,073
16.	$AE = v_{16}$	129,196
17.	$AF = v_{17}$	118,206
18.	$AG = v_{18}$	201,218

No.	Titik Asal	Nilai $h(n)$ (km)
19.	AH = v_{19}	218,855
20.	AI = v_{20}	148,049
21.	AJ = v_{21}	170,175
22.	AK = v_{22}	194,621
23.	AL = v_{23}	161,447
24.	AM = v_{24}	161,448
25.	AN = v_{25}	225,994
26.	AO = v_{26}	225,985
27.	AP = v_{27}	168,813
28.	AQ = v_{28}	153,785
29.	AR = v_{29}	198,965
30.	AS = v_{30}	202,497
31.	AT = v_{31}	195,312

Pencarian lintasan terpendek dari Dinas Pendidikan Provinsi Lampung ke 30 SMA Negeri di Provinsi Lampung dilakukan dengan 2 pengerjaan yaitu secara manual dan dengan bantuan *Python*. Karena perhitungan manual dengan menggunakan algoritme Dijkstra memerlukan tempat yang banyak, maka pada artikel in hanya ditampilkan perhitungan manual sampai iterasi ke 4 yang dapat dilihat pada Tabel 3.

Tabel 3. Penghitungan manual dengan algoritme Dijkstra sampai iterasi ke 4.

Iterasi	V	C	Q	S	T	U	V	W	X	Y	Z	AA	AB
0	{A}	2,4	13,1	24,3	49,2	41,7	54,6	46,2	48,1	66	∞	72,7	91,5
1	{A,C}	2,4	13,1	24,3	49,2	41,7	54,6	46,2	48,1	66	∞	72,7	95,7
2	{A,C,Q}	2,4	13,1	24,3	49,2	41,7	54,6	46,2	48,1	66	∞	72,7	95,7
3	{A,C,Q,S}	2,4	13,1	24,3	49,2	41,7	54,6	46,2	48,1	66	97,8	72,7	95,7
4	{A,C,Q,S,U}	2,4	13,1	24,3	49,2	41,7	54,6	46,2	48,1	66	97,8	72,7	95,7

Perhatikanlah nilai-nilai yang diberi warna merah merupakan jarak terpendek yang didapat dari titik A (Dinas Pendidikan propinsi Lampung) ke titik titik lainnya. Sebagai contoh, pada iterasi ke-2 didapat jarak terpendek dari Dinas Pendidikan Propinsi Lampung ke SMAN 1 Gedong Tataan sejauh 13,1 km dan pada iterasi ke-3 didapat jarak terpendek dari Dinas Pendidikan Propinsi Lampung ke SMAN 1 Pringsewu sejauh 41,7 km. Hal yang sama untuk penghitungan manual dengan algoritme A* hanya diberikan sampai tiga kali daftar O (*open*) dan C (*closed*) yang tersedia pada Tabel 4.

Tabel 4. Penghitungan manual dengan algoritme A* untuk 3 kali daftar O dan C.

No.	Titik	Daftar O	Daftar C	Titik Terpilih
1.	C,Q,S,T,U,V,W,X,Y,A A,AB,AD,AE,AF,AI,A K,AL,AM,AO,AQ	C,Q,S,T,U,V,W,X,Y,AA,AB,AD,AE,AF ,AI,AK,AL,AM,AO,AQ	A	C
2.	Q,S,T,U,V,W,X,Y,AA, AB,AD,AE,AF,AI,AK, AL,AM,AO,AQ	Q',S',T',U',V',W',X',Y',AA',AB',AD', AE',AF',AI',AK',AL',AM',AO',AQ',Q, S,T,U,V,W,X,Y,AA,AB,AD,AE,AF,AI, AK,AL,AM,AO,AQ	A,C	AA
3.	Q,S,T,U,V,W,X,Y,Z,A B,AD,AE,AF,AI,AK,A L,AM,AN,AO,AQ	Q',S',T',U',V',W',X',Y',AA',AB',AD', AE',AF',AI',AK',AL',AM',AO',AQ',Q '',S'',T'',U'',V'',W'',X'',Y'',A''A'',A B'',AD'',AE'',AF'',AI'',AK'',AL'',AM '',AO'',AQ'',Q,S,T,U,V,W,X,Y,Z,AB,A D,AE,AF,AI,AK,AL,AM,AN,AO,AQ,A B	A,C,AA	AB

Selain melakukan perhitungan manual, dilakukan juga penghitungan dengan menggunakan bahasa pemrograman *Python*. Pada Gambar 1 dan 2 berikut ini diberikan bagian dari *script* untuk kedua algoritme tersebut yang berisi input data titik-titik lokasi.

```
def jalur_dilewati(parents, start, end):
    jalur = [end]
    while True:
        key = parents[jalur[0]]
        jalur.insert(0, key)
        if key == start:
            break
    return jalur

input_simpul = ('A', 'C', 'Q', 'S', 'T', 'U', 'V', 'W', 'X',
               'Y', 'Z', 'AA', 'AB', 'AC', 'AD', 'AE', 'AF', 'AG', 'AH',
               'AI', 'AJ', 'AK', 'AL', 'AM', 'AN', 'AO', 'AP', 'AQ', 'AR',
               'AS', 'AT')

input_graf = {'A' : {'C': 2.4, 'Q': 13.1, 'S': 24.3, 'T':
49.2, 'U': 41.7, 'V': 54.6, 'W': 46.2, 'X': 48.1, 'Y': 66,
'AA': 72.7, 'AB': 91.5, 'AD': 98.1, 'AE': 97.1, 'AF': 72.3,
'AI': 113, 'AK': 131, 'AL': 162, 'AM': 123, 'AO': 222, 'AQ':
180),
```

Gambar 1. Bagian *script* algoritme Disjkstra.

```
def __init__(self, adjacency_list):
    self.adjacency_list = adjacency_list

def get_neighbors(self, v):
    return self.adjacency_list[v]

def h(self, n):
    H = {
        'A': 0,
        'C': 2.517,
        'Q': 9.301,
        'S': 18.085,
        'T': 33.875,
        'U': 33.566,
        'V': 36.255,
        'W': 37.790,
        'X': 39.413,
        'Y': 55.092,
        'Z': 74.759,
```

Gambar 2. Bagian *script* algoritme A*.

Solusi yang didapatkan pada perhitungan manual menghasilkan lintasan terpendek yang sama yaitu sebagai berikut:

1. Lintasan terpendek dari v_1 ke v_2 Dinas Pendidikan Provinsi Lampung-SMAN 2 Bandar Lampung adalah 2,4 km.
2. Lintasan terpendek dari v_1 ke v_3 yaitu Dinas Pendidikan Provinsi Lampung-SMAN 12 Bandar Lampung adalah 13,1 km.
3. Lintasan terpendek dari v_1 ke v_4 yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Gedong Tataan adalah 24,3 km.
4. Lintasan terpendek dari v_1 ke v_5 Dinas Pendidikan Provinsi Lampung-SMAN 1 Way Khilau adalah 49,2 km.
5. Lintasan terpendek dari v_1 ke v_6 yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Pringsewu adalah 41,7 km.
6. Lintasan terpendek dari v_1 ke v_7 yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Pardasuka adalah 54,6 km.
7. Lintasan terpendek dari v_1 ke v_8 yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Metro adalah 46,2 km.
8. Lintasan terpendek dari v_1 ke v_9 yaitu Dinas Pendidikan Provinsi Lampung-SMAN 5 Metro adalah 48,1 km.
9. Lintasan terpendek dari v_1 ke v_{10} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Gunung Sugih adalah 66 km.
10. Lintasan terpendek dari v_1 ke v_{11} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Metro-SMAN 1 Way Seputih adalah 95,30000000000001 km.
11. Lintasan terpendek dari v_1 ke v_{12} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 2 Bandar Lampung-SMAN 1 Kalianda adalah 62,6 km.
12. Lintasan terpendek dari v_1 ke v_{13} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 2 Bandar Lampung-SMAN 1 Kalianda-SMAN 1 Bakauheni adalah 91 km.
13. Lintasan terpendek dari v_1 ke v_{14} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Metro-SMAN 1 Sukadana adalah 74,7 km.
14. Lintasan terpendek dari v_1 ke v_{15} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Pasir Sakti adalah 98,1 km.
15. Lintasan terpendek dari v_1 ke v_{16} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Kotaagung adalah 97,1 km.
16. Lintasan terpendek dari v_1 ke v_{17} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Gunung Alip adalah 72,3 km.
17. Lintasan terpendek dari v_1 ke v_{18} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Gedong Tataan-SMAN 1 Bukit Kemuning- SMAN 1 Blambangan Umpu adalah 191,3 km.
18. Lintasan terpendek dari v_1 ke v_{19} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Kota Bumi-SMAN 1 Buay Bahuga adalah 212,4 km.
19. Lintasan terpendek dari v_1 ke v_{20} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Kota Bumi adalah 113 km.
20. Lintasan terpendek dari v_1 ke v_{21} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Gedong Tataan-SMAN 1 Bukit Kemuning adalah 140,3 km.
21. Lintasan terpendek dari v_1 ke v_{22} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Tulang Bawang Tengah adalah 131 km.
22. Lintasan terpendek dari v_1 ke v_{23} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Gunung Agung adalah 162 km.
23. Lintasan terpendek dari v_1 ke v_{24} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Menggala adalah 123 km.
24. Lintasan terpendek dari v_1 ke v_{25} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Menggala-SMAN 1 Penawartama adalah 189,3 km.
25. Lintasan terpendek dari v_1 ke v_{26} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Menggala-SMAN 1 Mesuji adalah 221,7 km.
26. Lintasan terpendek dari v_1 ke v_{27} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Menggala-SMAN 1 Penawartama-SMAN 1 Rawajitu Utara adalah 218,9 km.

27. Lintasan terpendek dari v_1 ke v_{28} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Bengkuntat adalah 180 km.
28. Lintasan terpendek dari v_1 ke v_{29} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Bengkuntat-SMAN 1 Pesisir Selatan adalah 221,4 km.
29. Lintasan terpendek dari v_1 ke v_{30} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Gedong Tataan-SMAN 1 Liwa adalah 204,3 km.
30. Lintasan terpendek dari v_1 ke v_{31} yaitu Dinas Pendidikan Provinsi Lampung-SMAN 1 Kotaagung-SMAN 1 Belalau adalah 208,1 km.

4. KESIMPULAN

Berdasarkan hasil dan pembahasan yang telah diuraikan dapat disimpulkan bahwa pencarian lintasan terpendek ke 30 SMA Negeri di Provinsi Lampung dengan titik awal Dinas Pendidikan Provinsi Lampung menggunakan algoritme Dijkstra dan algoritme *A-Star* secara manual maupun dengan bahasa pemrograman *Python* mendapatkan hasil yang sama secara keseluruhan ke 30 titik tujuan. Walaupun demikian, dari perbandingan waktu proses dengan menggunakan bahasa pemrograman *Python* dari kedua algoritme tersebut, didapat rata-rata *running time* algoritme Dijkstra adalah 78, 575 ms dan hasil rata-rata *running time* algoritme *A** adalah 47,505 ms. Dari nilai tersebut, dapat disimpulkan bahwa rata-rata *running time* algoritme *A** (*A-Star*) lebih cepat 31,7 ms dibandingkan dengan algoritme Dijkstra.

DAFTAR PUSTAKA

- [1] Mayadi, M. dan Azhar, R. Perbandingan Perhitungan Manual dengan Algoritma *A-Star* dalam Pencarian Jalur Terpendek untuk Pengiriman Pesanan Dodol Khas Lombok. *Jurnal Informatika dan Rekayasa Elektronik*. Vol. 2 (2), pp: 27-34. 2019.
- [2] Deo, N. *Graph Theory with Application to Engineering and Computer Science*. USA: Prentice-Hall, Inc. 1989.
- [3] Marcelina, D. dan Yulianti, E. Aplikasi Pencarian Rute Terpendek Lokasi Kuliner Khas Palembang menggunakan Algoritma Euclidean Distance dan *A*(Star)*. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*. Vol. 9 (2), pp: 195–202. 2020.
- [4] Schuerer, K. dan & Maufrais, C. *Introduction to Programming using Python*. Pearson, Boston. 2010.
- [5] Lutz, M. *Learning Python, Fourth Edition*. United States of America: O'Reilly Media. 2009.
- [6] Saputra, R. Sistem Informasi Pencarian Obyek Kota Medan dengan Algoritma Dijkstra. *Jurnal Matematika*. Vol 14 (1), pp: 19-24. 2011.
- [7] Cantona, A., Fauziah, F., dan Winarsih, W. Impelemantasi Algoritma Dijkstra pada Pencarian Rute Terpendek ke Museum di Jakarta. *Jurnal Teknologi dan Manajemen Informatika*. Vol. 6 (1), pp: 27–34. 2020.
- [8] Luthfita, D. dan Aripin, S. Implementasi Algoritma *A** dalam menentukan Tarif Minimum Berdasarkan Jarak Terpendek Rute Armada Taksi Bandara. *Journal of Informatics Management and Information Technology*. Vol. 2 (1), pp: 43–47. 2022.
- [9] Rizky, R. Pencarian Jalur Terdekat dengan Metode *A*(Star)* Studi Kasus Serang Labuan Provinsi Banten. *Prosiding Seminar Nasional Rekayasa Teknologi Informasi (SNARTISI)*, November 2018, pp: 93–98. 2018.
- [10] Zhang, Y., Li, L. L., Lin, H. C., Ma, Z., dan Zhao, J. Development of path planning approach using improved *A-star* algorithm in AGV system. *Journal of Internet Technology*. Vol 20 (3), pp: 915-924. 2019.