

Implementasi Algoritma Floyd Warshall dalam Pencarian Rute Terpendek Lokasi Tower Base Transceiver Station (BTS) pada PT Citra Akses Indonusa

Bella Nafa Savitri*, Mohammad Nafie Jauhari, Evawati Alisah, Hisyam Fahmi

Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang, Indonesia

bellanafasavitri@gmail.com, nafie.jauhari@mat.uin-malang.ac.id, evawatialisah@mat.uin-malang.ac.id, hisyam.fahmi@uin-malang.ac.id

Abstrak

PT Citra Akses Indonusa merupakan perusahaan yang berjalan pada keahlian layanan teknologi informasi di Provinsi Banten. Salah satu layanan yang disajikan oleh perusahaan tersebut membutuhkan pembangunan *tower Base Transceiver Station* (BTS). *Tower* tersebut membutuhkan pemeliharaan jika terdapat kerusakan pada sinyal jaringan. Akibatnya dibutuhkan rute terpendek untuk memudahkan karyawan menuju lokasi *tower* agar lebih efektif. Terdapat delapan lokasi *tower* BTS di Kabupaten Tangerang dan tiga lokasi *tower* BTS di Kota Tangerang. Proses pencarian rute terpendek pada penelitian ini menggunakan algoritma *Floyd Warshall*, yang memiliki keunikan mencari rute terpendek dengan membandingkan setiap sisi dari seluruh sisi yang dilewati. Proses pengujian rute terpendek dilakukan dengan memilih titik awal, kemudian memilih beberapa lokasi *tower* BTS. Selanjutnya akan dicari rute terpendek menggunakan algoritma *Floyd Warshall* dari setiap titik tujuan lokasi *tower* BTS, kemudian akan dipilih lokasi *tower* BTS yang akan dikunjungi terlebih dahulu begitu pun seterusnya hingga tujuan terakhir. Keefektifan pencarian rute terpendek ini melibatkan perbandingan rute yang disajikan oleh *Google Maps*. Berdasarkan hasil 30 kali percobaan secara acak terhadap lokasi *tower* BTS, didapatkan rata-rata efektivitas rute terpendek sebesar 25.54% dibandingkan rute yang dihasilkan *Google Maps*. Hal ini dikarenakan adanya penyeleksian pada tiap lokasi tujuan *tower* BTS sehingga menjadikan rute lebih efektif.

Kata kunci: Graf; rute terpendek; *Floyd Warshall*; *tower*; PT Citra Akses Indonusa

Abstract

PT Citra Akes Indonusa is a company that operates on the expertise of information technology services in Banten Province. One of the services provided by the company requires the construction of a Base Transceiver Station (BTS) tower. The tower requires maintenance if there is damage to the network signal. As a result, the shortest route is needed to make it easier for employees to reach the tower location to be more effective. There are eight BTS tower locations in Tangerang Regency and three BTS tower locations in Tangerang City. The process of finding the shortest route in this study uses the Floyd Warshall algorithm, which is unique in finding the shortest route by comparing each edge of all edges that are passed. The process of testing the shortest route is done by selecting the starting point, then selecting several BTS tower locations. Next, the shortest route will be searched using the Floyd Warshall algorithm from each point of destination for the BTS tower location, then the BTS tower location will be selected first and so on until the last destination. The effectiveness of this shortest route search involves a comparison of the routes presented by Google Maps. Based on the results of 30 randomized trials on BTS tower locations, the average shortest route effectiveness was 25.54% compared to the route generated by Google Maps. This is due to the selection at each BTS tower destination location so as to make the route more effective.

Keywords: Graf; shortest path; Floyd Warshal; *tower*; PT Citra Akses Indonusa

PENDAHULUAN

PT Citra Akses Indonusa merupakan salah satu perusahaan yang berjalan pada keahlian layanan teknologi informasi di Provinsi Banten yang didirikan pada tahun 2017. Salah satu layanan yang disajikan perusahaan tersebut merupakan layanan *networking*. Layanan *networking* membutuhkan pembangunan *tower Base Transceiver Station* (BTS) sebagai alat pemancar sinyal. BTS itu sendiri merupakan sebuah alat dalam jaringan telekomunikasi berupa *tower* yang memiliki suatu antena yang memancarkan sinyal dan berfungsi sebagai penguat sinyal daya yang menghubungkan antara jaringan operator telekomunikasi dengan para *customer* [1]. *Tower* tersebut juga membutuhkan *maintenance* atau pengaturan kembali jika terdapat kerusakan pada sinyal jaringan. Maka untuk memudahkan akses karyawan sampai ke lokasi BTS dan melakukan *maintenance* dengan optimal, dibutuhkan rute terpendek. Teori yang berkaitan erat dengan pencarian rute terdekat adalah teori tentang graf. Graf G merupakan pasangan $(V(G), E(G))$ dengan $V(G)$ merupakan himpunan tidak kosong dan berhingga dari objek-objek yang disebut titik dan $E(G)$ adalah himpunan (mungkin kosong) pasangan tak berurutan dari titik-titik berbeda di $V(G)$ yang disebut sisi [2]. Salah satu algoritma graf dalam pencarian rute terpendek adalah algoritma *Floyd Warshall*.

Algoritma *Floyd Warshall* adalah algoritma yang menerapkan pemrograman yang dinamis, yaitu metode penyelesaian permasalahan dengan melihat dari hasil yang hendak dicapai sebagai suatu kesimpulan yang saling berhubungan [3]. Dalam pemrosesan satu matriks digunakan rumus sebagai berikut [4]:

$$F_{[i,j]}^k = \min\{F_{[i,j]}^{k-1}, F_{[i,k]}^{k-1} + F_{[k,j]}^{k-1}\} \quad (1)$$

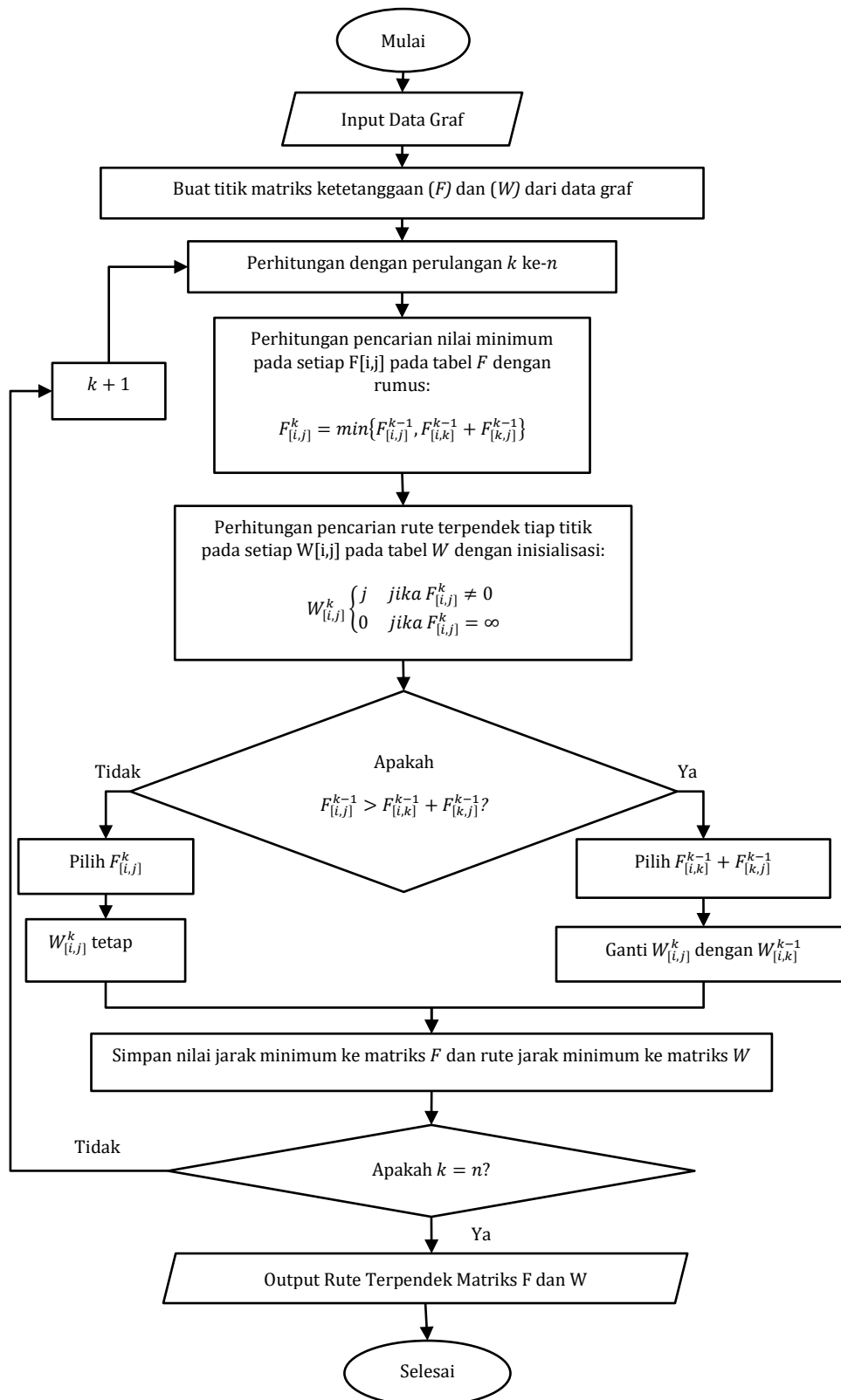
Di mana $F_{[i,j]}^k$ menyatakan entri matriks F ketetapan berbobot, baris ke- i , kolom ke- j di iterasi ke- k .

Rumus berikut hanya mendapatkan hasil nilai jarak terpendek dari tiap-tiap titik lokasi *tower* BTS. Diperlukan rumus modifikasi tambahan untuk menentukan jalur/rute terpendek, yang disusun sebagai berikut:

$$W_{[i,j]}^k = \begin{cases} j & \text{jika } F_{[i,j]}^k \neq 0 \\ 0 & \text{jika } F_{[i,j]}^k = \infty \end{cases} \quad (2)$$

Di mana $W_{[i,j]}^k$ menyatakan entri matriks W yang berisi simbol lokasi titik, baris ke- i , kolom ke- j di iterasi ke- k .

Berdasarkan modifikasi tersebut, ketika dalam iterasi ke- k nilai $F_{[i,j]}$ lebih besar dari $F_{[i,k]} + F_{[k,j]}$, maka akan dipilih nilai $F_{[i,k]} + F_{[k,j]}$ yang juga dilakukan dengan mengganti $W_{[i,j]}^k$ dengan $W_{[i,k]}^{k-1}$. Berikut merupakan gambar *flowchart* dari algoritma *Floyd Warshall*:



Gambar 1. Flowchart Algoritma Floyd Warshall

METODE

Data Penelitian

The Data pada penelitian ini menggunakan data sekunder lokasi *tower Base Transceiver Station* (BTS) yang diambil berdasarkan pengguna yang bekerja sama dengan PT Citra Akses Indonusa pada tahun 2020.

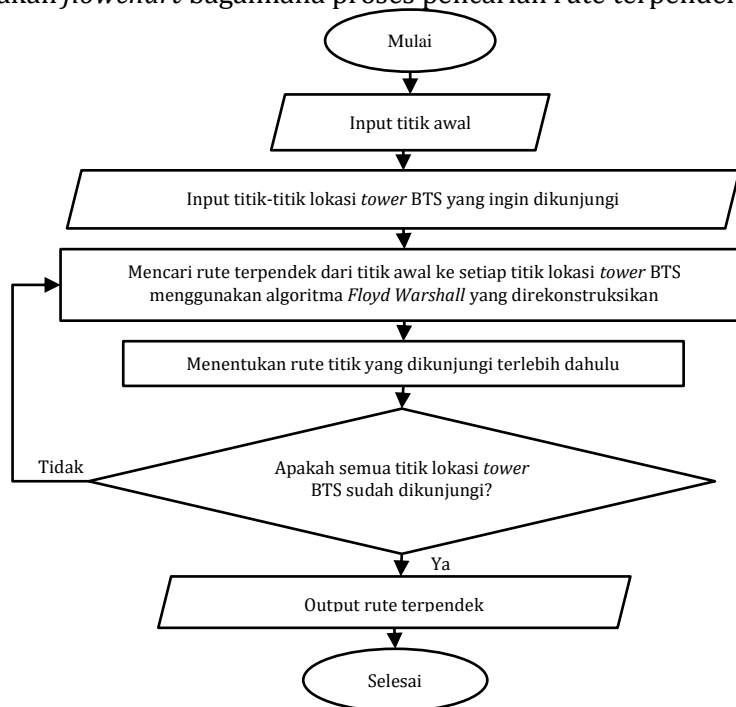


Gambar 2. Titik Lokasi *Tower* BTS

Langkah-Langkah Penelitian

Adapun langkah-langkah yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Peninjauan secara langsung objek yang akan digunakan sebagai bahan penelitian.
2. Pengumpulan data lokasi *tower Base Transceiver Station* (BTS) dan lokasi persimpangan yang dijadikan sebagai titik. Akan diambil rute yang dilewati secara umum atau rute yang melewati jalan raya dikarenakan rute ini memungkinkan untuk dilalui kendaraan roda 4.
3. Pembuatan graf lokasi *tower* BTS ini melibatkan penggunaan program *Python*.
4. Pengujian pencarian rute terpendek menggunakan algoritma *Floyd Warshall*. Pengujian yang dilakukan akan memasukkan titik awal dan beberapa titik tujuan lokasi *tower* BTS, kemudian akan dipilih rute mana yang lebih dekat terlebih dahulu dengan titik awal kemudian dicari titik tujuan berikutnya hingga ditemukan titik tujuan terakhir. Berikut ini merupakan *flowchart* bagaimana proses pencarian rute terpendek bekerja:



Gambar 3. Flowchart Pengujian Pencarian Rute Terpendek

5. Pengevaluasian hasil pengujian dari rute terpendek yang dihasilkan.

HASIL DAN PEMBAHASAN

Deskripsi Data

1. Data Titik Lokasi Tower BTS

PT Citra Akses Indonusa memiliki tower BTS yang tersebar di daerah Kabupaten dan Kota Tangerang. Penjabaran lokasi tower BTS yang berada di Kota Tangerang disajikan pada tabel 1:

Tabel 1. Lokasi tower BTS Kota Tangerang

No.	Nama Lokasi Tower BTS	Simbol
1	Novotel	NOVOTEL
2	Great Western Resort	GWR
3	Ronal (Selapanjang)	RS

Simbol tersebut akan digunakan agar pemilihan lokasi tower BTS yang diinginkan lebih mudah.

2. Data Titik Lokasi Tower BTS

Data titik lokasi persimpangan jalan diambil berdasarkan jenis jalan yang berdasarkan fungsi dari jalan raya memiliki arti jalan arteri primer, arteri sekunder, kolektor primer, kolektor sekunder, dan lokal primer. Pemilihan jenis jalan tersebut berlandaskan pada jalan umum atau jalan yang sering dilalui oleh pengguna berkendara roda empat. didapatkan titik lokasi persimpangan jalan sebanyak 218 titik dan 517 sisi yang akan dimuat ke dalam format data graf berarah seperti pada tabel 2. Tabel tersebut berisikan pasangan titik dan bobot dengan satuan kilometer.

Tabel 2. Penulisan Format Data Graf

Titik 1	Titik 2	Bobot (Km)
11	29	0.45

Program Pencarian Rute Terpendek

1. Program Pembacaan Data

Dibutuhkan suatu program pembacaan data untuk mengambil data dari format data pada tabel 2. Digunakan modul *networkx* sebagai *nx* untuk membantu pembacaan *file* data dan data tersebut akan dimuat ke dalam bentuk graf berarah. Berikut merupakan program fungsi pembacaan data:

```
g=nx.read_edgelist('nama_file',create_using=nx.DiGraph(),data=[('bobot',float)])
```

Gambar 4. Program Fungsi Pembacaan Data Graf

Program ini merupakan program dasar pembacaan data agar dapat dilakukan pencarian rute terpendek.

2. Program Algoritma Floyd Warshall

Dari data yang sudah terbaca pada tahapan pembacaan file data pada program Python akan didefinisikan algoritma Floyd Warshall dengan memasukkan fungsi graf dan bobot yang diketahui. Gambar 5 merupakan potongan gambar dari program algoritma *Floyd Warshall*:

```

for i, j, k in graph.edges(data=True):
    bobot = k.get(weight,1.0)
    jarak[i][j] = min(bobot, jarak[i][j])
    titik[i][j] = i

for w in graph:
    jarak_w = jarak[w] # simpan perhitungan
    for i in graph:
        jarak_i = jarak[i] # simpan perhitungan
        for j in graph:
            k = jarak_i[w] + jarak_w[j]
            if jarak_i[j] > k:
                jarak_i[j] = k
                titik[i][j] = titik[w][j]

return dict(titik), dict(jarak)

```

Gambar 5. Potongan Program Algoritma Floyd Warshall

Dari pendefinisian algoritma tersebut, didapatkan hasil rute terpendek dan bobot dari seluruh rute yang dilalui dari tiap titik. Pada tabel 3 akan ditampilkan hasil rute terpendek yang berisikan bobot dari tiap-tiap titik tower BTS.

Tabel 3. Hasil Bobot Rute Terpendek Tiap-Tiap Lokasi Titik Tower BTS

TITI K	CAI	AR I	SS B	G W R	BM W	RS	VB	PD KL	DE NI	RO NA L	NOV OTE L
CAI	0	8.8 2	16. 84	17. 6	26. 69	15. 28	18. 18	20. 36	7.1	10. 5	14.1 7
ARI	10. 12	0	24. 95	16. 17	26. 17	6.5 8	26. 06	21. 16	7.1 4	1.8 1	12.2 5
SSB	16. 79	25. 13	0	21. 96	26. 95	29. 59	14. 51	19. 8	22. 45	26. 81	23.4 3
GW R	16. 09	15. 44	22. 68	0	10. 79	13. 39	25. 74	10. 32	19. 92	15. 27	4.5
BM W	25. 43	24. 72	26. 75	9.7 4	0	23	31. 44	7.0 5	29. 2	24. 86	14.0 9
RS	16. 58	6.5 8	29. 22	13. 7	23. 69	0	31. 11	22. 71	12. 2	4.7 7	9.78
VB	17. 25	25. 6	13. 98	24. 97	31. 73	31. 11	0	24. 58	20. 66	26. 35	27.8 5
PDK L	20. 57	21. 85	20. 05	10. 22	7.1 5	21. 13	25. 36	0	25. 22	22. 96	12.2 7
DEN I	7.1	7.1 4	22. 5	21. 18	31. 19	12. 2	20. 66	24. 97	0	7.4 3	17.2 7
RON AL	11. 80	1.8 1	26. 63	16. 63	26. 63	4.7 7	26. 35	22. 84	7.4 3	0	12.7 1
NOV OTE L	15. 09	12. 5	24. 34	4.5	14. 5	10. 09	28. 89	13. 62	17	11. 96	0

3. Program Rekonstruksi jalur

Program algoritma Floyd Warshall diketahui hanya mencari seluruh jalur terpendek dari setiap titik-titik. Dibutuhkan program tambahan untuk mencari pencarian dari satu titik awal menuju satu titik tujuan. Berikut merupakan gambar dari program rekonstruksi jalur sebagai program tambahan pencarian jalur:

```
#Bentuk pencarian rute dengan dimulai titik awal hingga titik tujuan
def rekonstruksi_jalur(t_awal, t_tujuan, predecessors):
    if t_awal == t_tujuan:
        return []
    t_sebelum = predecessors[t_awal]
    t_sekarang = t_sebelum[t_tujuan]
    jalur = [t_tujuan, t_sekarang]
    while t_sekarang != t_awal:
        t_sekarang = t_sebelum[t_sekarang]
        jalur.append(t_sekarang)
    return list(reversed(jalur))
```

Gambar 6. Pendefinisian Program Rekonstruksi Jalur

4. Program Pencarian Rute Terpendek Terlebih Dahulu

Program ini akan dibuat suatu pencarian rute terpendek dari satu titik awal menuju beberapa titik tujuan. Sesuai dengan nama pendefinisian dari beberapa titik tujuan tersebut akan dipilih rute paling terpendeknya diikuti dengan rute terpendek selanjutnya. Gambar 7 menampilkan potongan gambar program rute terpendek terlebih dahulu.

```
else :
    for i in titiktujuan:
        lintasan=rekonstruksi_jalur(X,i,semuajalur)
        jarak=semuajarak[X][i]
        penyeleksian.append([lintasan, jarak])

    A1=min(penyeleksian, key=lambda x:x[1] )
    penyeleksian.clear()
    rute.append(A1[0])
    jarak1=A1[1]
    totaljarak+=jarak1
    A2=(A1[0][-1])
    titiktujuan.remove(A2)
    return penyeleksianrtd(A2, titiktujuan, rute, jarak, totaljarak)

penyeleksianrtd(X, titiktujuan, rute, jarak, totaljarak)
```

Gambar 7. Potongan Program Rute Terpendek Terlebih Dahulu

Pengujian Pencarian Rute Terpendek

Proses pengujian pencarian rute terpendek dilakukan sesuai dengan tahapan penelitian algoritma *Floyd Warshall* yang telah dijelaskan pada bab 3.

1. Menentukan Titik Awal dan Titik Tujuan

Pada tahapan ini titik awal dan titik tujuan dapat dipilih secara bebas sesuai dengan titik lokasi tower BTS yang terdapat pada data yang ada yang dapat dimasukkan beberapa tujuan titik lokasi tower BTS dengan tidak memasukkan titik lebih dari data yang ada. Format penulisan untuk memasukkan beberapa titik tujuan pada program dipisahkan dengan tanda koma (,). Contoh format memasukkan titik awal dengan beberapa titik tujuan disajikan pada tabel 4.

Tabel 4. Contoh Format Penulisan Memasukkan Titik Awal dan Titik Tujuan

Titik Awal	CAI
Titik Tujuan	RONAL,ARI,RS

2. Pengujian Rute Terpendek

Tahapan pengujian rute terpendek ini akan diseleksi titik awal terhadap semua lokasi titik tujuan yang dipilih menggunakan algoritma *Floyd Warshall*. Berikut merupakan contoh pengujian rute terpendek berdasarkan contoh titik pada tabel 4. Pada pengujian penyeleksian rute terdekat terlebih dahulu ini akan dicari rute tiap-tiap titik awal ke dalam daftar list tutup. Apabila sudah ditemukan rute terpendeknya, maka akan dimasukkan ke

dalam daftar list rute.

Penentuan Rute CAI → RONAL

Berdasarkan hasil dari penyelesaian pencarian rute terpendek menggunakan algoritma *Floyd Warshall*, diketahui bahwa bobot dari titik CAI menuju titik RONAL yaitu 10.50 Km. Proses penentuan ini diawali dengan memasukkan titik RONAL ke dalam daftar list tutup.

$$List\ tutup = ('RONAL')$$

Dari titik CAI menuju titik RONAL, titik yang dilalui adalah titik 14. Kita masukkan titik 14 tersebut ke dalam list tutup.

$$List\ tutup = ('RONAL', '14')$$

Selanjutnya titik yang dituju dari titik CAI menuju titik 14 adalah titik 58, masukkan ke dalam list.

$$List\ tutup = ('RONAL', '14', '58')$$

Kemudian dari titik CAI, titik yang dilalui menuju titik 58 adalah titik 12. Masukkan titik 12 tersebut ke dalam list tutup.

$$List\ tutup = ('RONAL', '14', '58', '12')$$

Sama dengan proses sebelumnya, dilakukan pencarian rute hingga titik yang dilalui untuk menuju titik CAI dari titik 1 adalah langsung titik CAI. Maka telah ditemukan rute terpendeknya dengan memasukkan titik CAI ke dalam list.

$$List\ tutup = ('RONAL', '14', '58', '12', '11', '10', '9', '3', '1', 'CAI')$$

Selanjutnya akan dicerminkan list tutup tersebut untuk membuat rute dari titik awal menuju titik tujuan sesuai dengan yang dicari. Didapatkan hasil yaitu:

$$List\ tutup = ('CAI', '1', '3', '9', '10', '11', '12', '58', '14', 'RONAL')$$

Proses ini dilakukan secara terus menerus hingga semua titik tujuan lokasi tower BTS telah dikunjungi seperti yang disajikan pada tabel 5.

Tabel 5. Hasil Penyeleksian Contoh Pengujian Rute

No	Penyeleksian	List Tutup	Bobot
1	CAI → RONAL	$List\ tutup = ('CAI', '1', '3', '9', '10', '11', '12', '58', '14', '12')$	10.5
	CAI → ARI	$List\ tutup = ('CAI', '1', '3', '9', '10', '11', '12', '58', 'ARI')$	8.82
	CAI → RS	$List\ tutup = ('CAI', '1', '3', '9', '10', '11', '12', '58', '14', 'RONAL', '25', '26', '27', '28', '38', 'RS')$	15.28
2	ARI → RONAL	$List\ tutup = ('ARI', '58', '14', 'RONAL')$	1.81
	ARI → RS	$List\ tutup = ('ARI', '58', '14', 'RONAL', '25', '26', '27', '28', '38', 'RS')$	6.58
3	RONAL → RS	$List\ tutup = ('RONAL', '25', '26', '27', '28', '38', 'RS')$	4.77

Dari seluruh penentuan rute yang telah dilakukan, didapatkan hasil list rute yang di mana telah dilalui tiap-tiap titik tujuan yang dimasukkan:

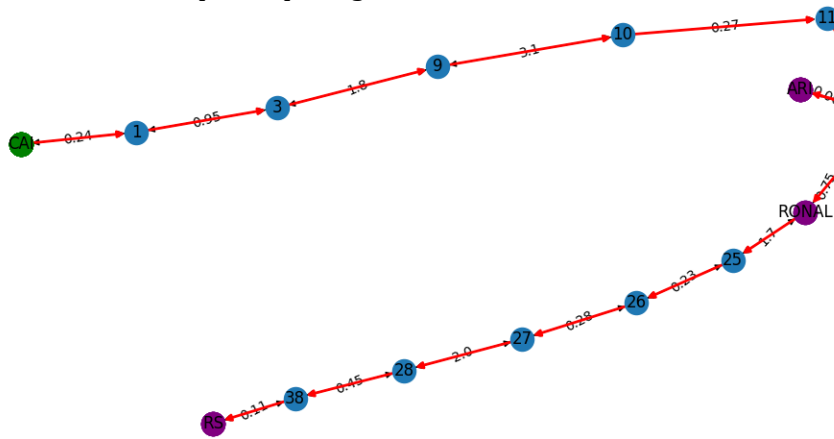
$$List\ rute = ('CAI', '1', '3', '9', '10', '11', '12', '58', 'ARI', '58', '14', 'RONAL', '25', '26', '27', '28', '38', 'RS')$$

List rute di atas menjelaskan bahwa rute yang dilalui pertama kali merupakan rute yang memiliki bobot yang paling terkecil dan dilanjutkan hingga semua titik tujuan sudah dilalui. Titik tujuan yang dituju pertama kali dari titik awal CAI adalah titik ARI. Kemudian dari titik ARI dilanjutkan menuju titik RONAL dan diakhiri dengan titik RS.

Total jarak dari seluruh rute yang dilalui adalah 15.40 Km. Apabila titik tujuan tersebut dimasukkan secara acak, akan tetap menghasilkan hasil yang sama.

3. Visualisasi Rute

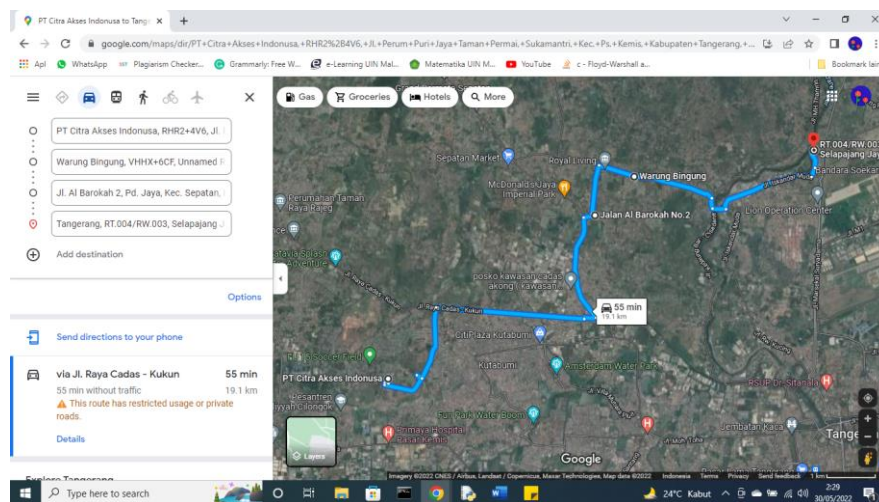
Hasil dari pencarian rute dari titik awal CAI hingga titik tujuan akhir RS di visualisasikan dan ditampilkan pada gambar 8.



Gambar 8. Visualisasi Hasil Pengujian Rute

Hasil Pengujian

Penelitian ini menguji rute terpendek pada beberapa lokasi tower BTS menggunakan algoritma *Floyd Warshall* yang akan dikaji dengan *Google Maps*. Pada hasil pengujian pencarian rute sebelumnya didapatkan hasil rute yang diawali dengan CAI menuju ARI, RONAL dan RS. Pada *Google Maps* akan dilakukan pencarian rute yang sama sesuai urutan titik tujuan yang dimasukkan pada tabel 5.



Gambar 9. Hasil Pencarian Rute *Google Maps*

Gambar 9 menampilkan rute yang berurutan sesuai dengan titik tujuan yang dimasukkan. Didapatkan total jarak 19.1 Km yang lebih panjang 3.37 Km dibandingkan hasil pencarian menggunakan algoritma *Floyd Warshall*.

Proses pengujian ini akan dilakukan dengan acak secara menyeluruh sebanyak 30 kali. Jadi, seluruh titik lokasi tower BTS akan mendapatkan bagian untuk dilakukan proses pengujian sesuai dengan program yang telah dibuat sebelumnya. Akan dibandingkan hasil dari titik awal dan urutan titik tujuan yang dimasukkan menggunakan algoritma *Floyd Warshall* dan *Google Maps*. Hasil proses pengujian rute terpendek ini akan ditampilkan pada tabel 6.

Tabel 6. Hasil Pengujian Rute

No	Titik Awal	Titik Tujuan	Hasil Rute Penelitian	Hasil Jarak Penelitian (Km)	Rute Google Maps	Jarak Google Maps (Km)
1	SSB	ARI,GWR	SSB,GWR,ARI	37.40	SSB,ARI,GWR	41.8
2	NOVOTEL	RONAL,RS	NOVOTEL,RS,RONAL	14.86	NOVOTEL,RS,RONAL	16.5
3	VB	CAI,GWR	VB,CAI,GWR	34.86	VB,CAI,GWR	41.6
4	RONAL	BMW,CAI	RONAL,CAI,BMW	38.4	RONAL,BMW,CAI	53
...
30	CAI	NOVOTEL,SSB,DENI,GWR	CAI,DENI,NOVOTEL,GWR,SSB	51.56	CAI,NOVOTEL,SSB,DENI,GWR	89.1

Evaluasi Pengujian

Setelah melakukan pengujian sebanyak 30 kali didapatkan hasil bahwa pengujian menggunakan algoritma *Floyd Warshall* memiliki hasil jarak yang lebih pendek daripada jarak yang *Google Maps*. Efektivitas hasil dari jarak tempuh akan dimuat ke dalam bentuk persentase menggunakan rumus [5]:

$$\text{Nilai efektivitas} = \frac{JG - JF}{JG} \times 100\%. \quad (3)$$

Keterangan:

JG = jarak rute pada Google Maps

JF = jarak rute terpendek algoritma Floyd Warshall

Hasil dari perhitungan nilai efektivitas ditampilkan pada tabel 7.

Tabel 7. Hasil Perhitungan Nilai Efektivitas

Testing Number-	Percentage (%)
1	10,53
2	9,94
3	16,20
4	27,55
...	...
30	42,13
Rata-rata	25,54

Tabel 7 menunjukkan bahwa pengujian dengan algoritma *Floyd Warshall* memiliki jarak tempuh dengan rata-rata persentase keefektifan sebesar 25,54 % dibandingkan dengan *Google Maps*. Persentase keefektifan terbesar terjadi pada pengujian ke-28 sebesar 48,66%. Persentase keefektifan terkecil terjadi pada pengujian ke-7 sebesar 1,77%. Hasil persentase tersebut menyatakan bahwa algoritma *Floyd Warshall* merupakan algoritma yang efektif dikarenakan memiliki jarak yang lebih pendek dibandingkan dengan pengujian menggunakan *Google Maps*.

Alasan yang membuat hasil pengujian rute terpendek tersebut menjadi efektif adalah penyeleksian rute lokasi tower BTS yang dikunjungi di prioritaskan ke rute yang lebih dekat jaraknya terlebih dahulu dari seluruh tujuan yang dipilih. Apabila dimasukkan dua atau lebih titik lokasi tower BTS yang akan dituju, maka akan diseleksi setiap titik tujuan tersebut menggunakan program pencarian rute terpendek terlebih dahulu yang disesuaikan dengan algoritma *Floyd Warshall* yang direkonstruksikan. Selanjutnya akan dipilih titik lokasi tower BTS mana yang akan dijadikan tujuan pertama dan berlanjut hingga ditemukan titik tujuan terakhir. Dari proses pendefinisian program tersebut didapatkan pengujian rute yang lebih efektif dibandingkan dengan pengujian rute menggunakan *Google Maps* dikarenakan *Google Maps* tidak menyeleksi titik tujuan mana yang harus dituju terlebih dahulu.

KESIMPULAN

Dari hasil pengujian pencarian rute secara acak sebanyak 30 kali terhadap titik-titik lokasi tower BTS. Didapatkan rata-rata efektivitas jarak rute terpendek sebesar 25.54% dibandingkan jarak rute yang dihasilkan *Google Maps*. Hasil efektivitas tertinggi diperoleh pengujian ke-28 sebesar 48,66%. Persentase keefektifan terkecil terjadi karena memiliki hasil rute yang sama dengan *Google Maps*, yaitu pada pengujian ke-7 sebesar 1,77% yang berarti pencarian menggunakan algoritma *Floyd Warshall* memiliki hasil yang lebih efektif. Keefektifan dari hasil pencarian rute terpendek pada penelitian ini terjadi karena adanya program pencarian rute terpendek terlebih dahulu dengan tujuan untuk menentukan titik tujuan lokasi tower BTS yang memiliki jarak terdekat dengan titik awal untuk dikunjungi terlebih dahulu dari beberapa titik tujuan yang dipilih.

DAFTAR PUSTAKA

- [1] Nanang Ismail, Maharoni & Innel Lindra; "Analisis Perencanaan Pembangunan BTS (Base Transceiver Station) Berdasarkan Faktor Kelengkungan Bumi dan Daerah Fresnel di Regional Project Sumatera Bagian Selatan," *Jurnal Teknik Elektro*, vol. 09, no. 01, pp. 104-121, 2015.
- [2] Chartrand L. dan Lesniak L., *Graph & Digraph*, California: Chapman and Hall, 2000.
- [3] R. Saputra, "Sistem Geografis Pencarian Rute Optimum Obyek Wisata Kota Yogyakarta Dengan Algoritma Floyd-Warshall," *Jurnal Matematika*, vol. 14, no. 01, pp. 19-24, 2011.
- [4] L. Aldina, "Contoh Penerapan Program Dinamis dalam Algoritma Floyd-Warshall," Makalah: Sekolah Teknik Informatika Institut Teknologi Bandung, Bandung, 2018.
- [5] Mukti, M. Ridwan & Mulyono, "Menentukan Rute Terpendek Menggunakan Algoritma Floyd Warshall Dalam Pendistribusian Barang Pada PT. Rapy Ray Putratama," *Karismatika*, vol. 04, no. 01, pp. 39-53, 2018.
- [6] Aronoff, *Geographic Information System: A Management Perspective*, Canada, Ottawa: WDL Publication, 1989.
- [7] Laila Nugraha, A., & Sudarsono, B. , "Survei Topografi Untuk Menentukan Garis Tampak Pandang Base Transceiver Station (BTS)," *Teknik*, vol. 28, no. 1, pp. 55-60, 2012.
- [8] D. Isnaini, "Pencarian Rute Terpendek Non-Player Character (NPC) dengan Metode Floyd Warshall pada Game Wisata Batu," Universitas Islam Negeri Malang, Malang, 2019.
- [9] K. Surendro, "Pemanfaatan Enterprise Architecture Planning Untuk Perencanaan Strategis Sistem informasi," *Jurnal Informatika*, vol. 08, no. 01, pp. 1-9, 2007.
- [10] V. A. Nawagusti, "Penerapan Algoritma Floyd Warshall dalam Aplikasi Penentuan Rute Terpendek Mencari Lokasi BTS (Base Tower Station) pada PT.GCI Palembang," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 04, no. 02, pp. 081-088, 2018.

