

Optimalisasi Jaringan Jalan Antar Kecamatan dengan Minimum Spanning Tree dan Algoritma Prim di Kabupaten Ngawi

Amalia Ahsanti¹, Asyanada Insyafilla¹, Nadhifa Nur Fatimah¹, Winda Cahya Dwi Wahyuni¹, Deddy Rahmadi^{1,*}

¹ Program Studi Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga Yogyakarta

Dikirim: Desember 2024; Diterima: Maret 2025; Dipublikasi: Maret 2025

Alamat Email Korespondensi: deddy.rahmadi@uin-suka.ac.id

Abstrak

Penelitian ini mengoptimalkan jaringan jalan antar kecamatan di Kabupaten Ngawi dengan pendekatan Minimum Spanning Tree (MST) menggunakan Algoritma Prim untuk menentukan jalur terpendek. Jaringan jalan direpresentasikan sebagai graf berbobot, memungkinkan pemilihan jalur minimum tanpa siklus. Dengan menerapkan Algoritma Prim, diperoleh pohon merentang minimum dengan total jarak 146 km yang menghubungkan 19 kecamatan di kabupaten Ngawi secara efisien. Hasil penelitian menunjukkan bahwa pendekatan ini dapat menghasilkan rekomendasi optimalisasi jaringan transportasi yang lebih efisien. Implementasi MST berkontribusi pada penghematan biaya dan peningkatan aksesibilitas, mendukung distribusi barang dan layanan secara merata di Kabupaten Ngawi.

Kata Kunci:

Algoritma Prim, Jalur Terpendek, Ngawi

PENDAHULUAN

Di tengah pertumbuhan penduduk dan teknologi yang semakin berkembang pesat seperti saat ini, aksesibilitas antar wilayah perlu ditingkatkan. Aksesibilitas yang baik akan mendukung pemerataan pembangunan, optimalisasi jaringan transportasi, kemudahan akses, bahkan dasar bagi sistem navigasi dan logistik dalam kota.

Kabupaten Ngawi terdiri dari banyak kecamatan dan berbagai aktivitas ekonomi, sosial, dan pelayanan publik. Seringkali jalan yang tersedia saat ini tidak memaksimalkan jalur terpendek antar kecamatan, sehingga menyebabkan penggunaan sumber daya yang kurang efisien. Tanpa adanya perencanaan yang optimal, biaya pemeliharaan dan waktu tempuh dapat meningkat secara signifikan seiring berjalannya waktu. Oleh karena itu, diperlukan optimalisasi jaringan jalan dengan menentukan jalur terpendek antar kecamatan. Persoalan jalur terpendek biasanya di representasikan dalam bentuk graf. Graf merupakan kumpulan dari simpul (*vertex*) dan sisi (*edge*) [1]. Dalam hal ini, graf yang digunakan adalah graf berbobot (*weighted graph*). Graf berbobot adalah graf yang setiap busurnya diberi sebuah nilai atau bobot [2]. Bobot pada graf berbobot dapat menyatakan jarak antara dua kota, biaya perjalanan antara dua kota, biaya produksi, dan sebagainya. Sedangkan, titik atau simpul pada graf dapat menyatakan kecamatan di daerah tersebut.

Minimum Spanning Tree (MST) adalah himpunan bagian dari himpunan sisi (*edge*) suatu graf berbobot tak berarah yang menghubungkan semua simpul tanpa membentuk siklus dan dengan total bobot minimum [3]. Untuk mengoptimalkan aplikasi dari *Minimum Spanning Tree* (MST), diperlukan pula algoritma yang optimal. Oleh karena itu, pada penelitian ini akan digunakan Algoritma Prim untuk mencari *Minimum Spanning Tree* (MST). Algoritma Prim adalah algoritma dalam teori graf untuk mencari pohon rentang minimal untuk graf berbobot yang saling terhubung [4]. Algoritma Prim ditemukan oleh Ceko Vojtech Jarnik pada tahun 1930, kemudian ditemukan kembali pada tahun 1957 oleh Robert C. Prim, dan ditemukan kembali oleh Edsger W. Dijkstra pada tahun 1959.

Kajian menggunakan konsep Algoritma Prim telah banyak diterapkan dalam berbagai kasus, seperti pemasangan kabel [5], lokasi pembangunan jalan [1], penentuan rute di Google Maps [6], distribusi naskah soal [7], dan lain-lain [8-10]. Dalam artikel ini, akan ditentukan lintasan terpendek yang menghubungkan antar kecamatan di Kabupaten Ngawi menggunakan *Minimum Spanning Tree* dengan Algoritma Prim.

METODE PENELITIAN

1. Jenis dan Sumber Data

Data yang digunakan dalam penelitian ini adalah data Jarak Antar Kecamatan di Kabupaten Ngawi tahun 2023 yang diperoleh dari Badan Pusat Statistik (BPS). Data yang terdapat dalam penelitian ini adalah data jarak antar 19 kecamatan di Kabupaten Ngawi berdasarkan jarak per km.

2. Alat Penelitian

Alat penelitian yang digunakan dalam penelitian ini yaitu *software Visual Studio Code* dengan bahasa Phyton.

3. Variabel Penelitian

Variabel yang digunakan dalam penelitian ini adalah

A	: Kecamatan Sine	K	: Kecamatan Padas
B	: Kecamatan Ngrambe	L	: Kecamatan Kasreman
C	: Kecamatan Jogorogo	M	: Kecamatan Ngawi
D	: Kecamatan Kendal	N	: Kecamatan Paron
E	: Kecamatan Geneng	O	: Kecamatan Kedunggalar
F	: Kecamatan Gerih	P	: Kecamatan Pitu
G	: Kecamatan Kwadungan	Q	: Kecamatan Widodaren
H	: Kecamatan Pangkur	R	: Kecamatan Mantingan
I	: Kecamatan Karangjati	S	: Kecamatan Karanganyar
J	: Kecamatan Bringin		

4. Adapun langkah-langkah dalam penelitian ini adalah sebagai berikut:

1. Mempersiapkan data jarak antar kecamatan di Kabupaten Ngawi tahun 2023.
2. Membentuk graf berbobot jarak antar kecamatan di Kabupaten Ngawi tahun 2023.
3. Mencari *Minimum Spanning Tree* (MST) berdasarkan graf berbobot jarak antar kecamatan di Kabupaten Ngawi tahun 2023 menggunakan Algoritma Prim.
4. Menarik kesimpulan jarak terdekat antar kecamatan di Kabupaten Ngawi.

HASIL DAN PEMBAHASAN

Penelitian ini diawali dengan pengumpulan data jarak antar kecamatan di Kabupaten Ngawi tahun 2023 dengan sumber dari Badan Pusat Statistik (BPS). Pada

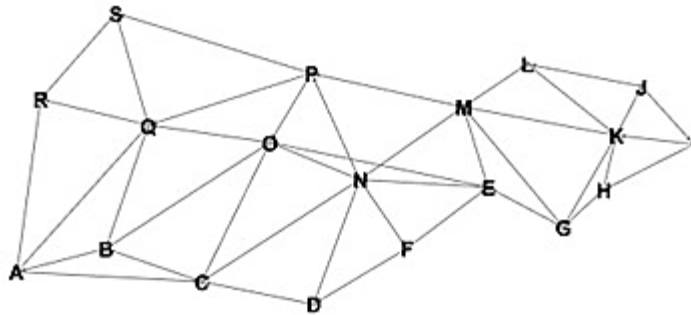
analisis ini, data akan disajikan dalam bentuk tabel yang mencakup semua kecamatan di wilayah Kabupaten Ngawi. Berikut adalah data jarak antar kecamatan yang dapat dilihat pada Tabel 1.

Tabel 1. Jarak Antar Kecamatan di Kabupaten Ngawi Tahun 2023

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
A	-	6	12	19	42	31	57	52	56	53	47	44	36	28	39	53	21	27	24
B	6	-	6	13	36	25	51	46	50	47	41	38	30	24	33	47	15	21	18
C	12	6	-	7	30	19	45	40	44	41	35	32	24	16	16	41	54	60	57
D	19	13	7	-	23	12	39	34	52	49	43	40	32	24	23	49	62	68	65
E	42	36	30	23	-	8	16	28	32	29	23	20	12	11	24	29	42	48	45
F	31	25	19	12	8	-	24	36	37	37	31	28	20	8	21	37	50	56	53
G	57	51	45	39	16	24	-	5	17	20	22	29	21	29	33	38	51	57	54
H	52	46	40	34	28	36	5	-	4	13	10	10	16	24	28	33	46	52	49
I	56	50	44	52	32	37	17	4	-	3	9	16	20	28	32	37	50	56	53
J	53	47	41	49	29	37	20	13	3	-	6	13	17	25	29	34	47	53	50
K	47	41	35	43	23	31	22	10	9	6	-	7	11	19	23	28	41	47	44
L	44	38	32	40	20	28	29	10	16	13	7	-	8	14	24	25	38	44	41
M	36	30	24	32	12	20	21	16	20	17	11	8	-	6	12	17	30	36	33
N	28	24	16	24	11	8	29	24	28	25	19	14	6	-	20	25	38	44	41
O	39	33	16	23	24	21	33	28	32	29	23	24	12	20	-	29	18	24	21
P	53	47	41	49	29	37	38	33	37	34	28	25	17	25	29	-	47	53	50
Q	21	15	54	62	42	50	51	46	50	47	41	38	30	38	18	47	-	6	3
R	27	21	60	68	48	56	57	52	56	53	47	44	36	44	24	53	6	-	9
S	24	18	57	65	45	53	54	49	53	50	44	41	33	41	21	50	3	9	-

Berdasarkan data di atas dilakukan pengolahan data dengan langkah – langkah :

1. Didaftar 19 kecamatan yang ada di Kabupaten Ngawi, yaitu Sine (A), Ngrambe (B), Jogorogo (C), Kendal (D), Geneng (E), Gerih (F), Kwadungan (G), Pangkur (H), Karangjati (I), Bringin (J), Padas (K), Kasreman (L), Ngawi (M), Paron (N), Kedunggalar (O), Pitu (P), Widodaren (Q), Mantingan (R), dan Karanganyar (S).
2. Representasi 19 kecamatan dalam bentuk graf.



Gambar 1. Representasi Kecamatan Ngawi dalam Bentuk Graf

3. Membuat *Minimum Spanning Tree* dengan algoritma prim. Menggunakan *software* Visual Studio Code dengan bahasa Python, akan diimplementasikan algoritma prim untuk mencari *Minimum Spanning Tree* dengan kode sebagai berikut:
 - a. Input: Memasukkan kode perintah algoritma prim untuk mencari *Minimum Spanning Tree* dan data jarak antar kecamatan di Kabupaten Ngawi.

```

1 import heapq
2
3 def prim(graph, start):
4     mst = []
5     visited = set([start])
6     edges = [
7         (cost, start, to) for to, cost in graph[start].items()
8     ]
9     heapq.heapify(edges)
10
11     while edges:
12         cost, frm, to = heapq.heappop(edges)
13         if to not in visited:
14             visited.add(to)
15             mst.append((frm, to, cost))
16
17             for to_next, cost in graph[to].items():
18                 if to_next not in visited:
19                     heapq.heappush(edges, (cost, to, to_next))
20
21     return mst

```

```

23 graph = {
24     'A': {'B': 6, 'C': 12, 'Q': 21, 'R': 27},
25     'B': {'A': 6, 'C': 6, 'O': 33, 'Q': 21},
26     'C': {'A': 12, 'B': 6, 'D': 7, 'N': 16, 'O': 16},
27     'D': {'C': 7, 'F': 12, 'N': 24},
28     'E': {'F': 8, 'M': 12, 'N': 11, 'O': 24},
29     'F': {'D': 12, 'E': 8, 'N': 8},
30     'G': {'E': 24, 'H': 5, 'K': 22, 'M': 21},
31     'H': {'G': 5, 'I': 4, 'K': 10},
32     'I': {'H': 4, 'J': 3, 'K': 9},
33     'J': {'I': 3, 'K': 6, 'L': 13},
34     'K': {'G': 22, 'H': 10, 'I': 9, 'J': 6, 'L': 7, 'M': 11},
35     'L': {'J': 13, 'K': 7, 'M': 8},
36     'M': {'E': 12, 'G': 21, 'K': 11, 'L': 8, 'N': 6, 'P': 17},
37     'N': {'C': 16, 'D': 24, 'E': 11, 'F': 8, 'M': 6, 'O': 20, 'P': 25},
38     'O': {'B': 33, 'C': 16, 'N': 20, 'P': 29, 'Q': 18},
39     'P': {'M': 17, 'N': 25, 'O': 29, 'Q': 47, 'S': 50},
40     'Q': {'A': 21, 'R': 6, 'S': 3},
41     'R': {'A': 27, 'Q': 6, 'S': 9},
42     'S': {'P': 50, 'Q': 3, 'R': 9}
43 }
44
45 mst = prim(graph, 'M')
46 print("Minimum Spanning Tree:", mst)

```

b. Output: Dari proses input di atas, diperoleh jarak terpendek antar kecamatan di Kabupaten Ngawi.

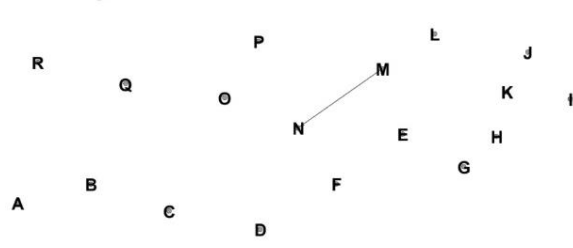
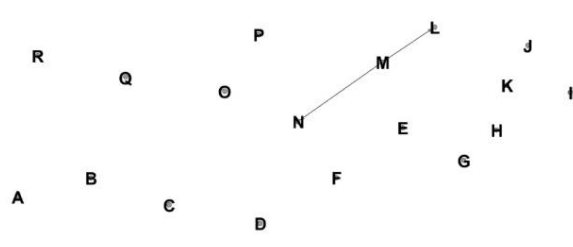
```

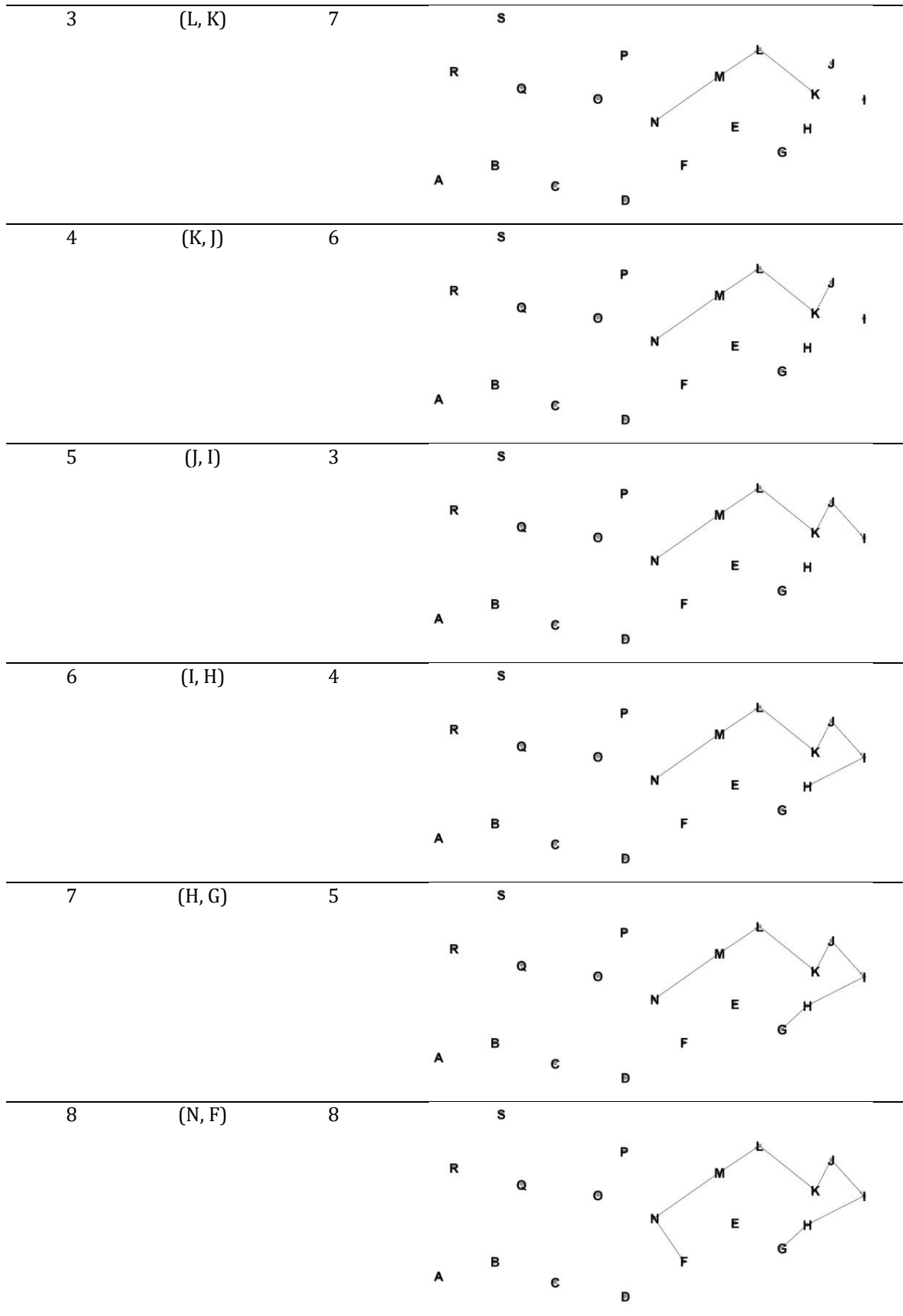
Minimum Spanning Tree: [(M, 'N', 6), (M, 'L', 8), (L, 'K', 7), (K, 'J', 6), (J, 'I',
3), (I, 'H', 4), (H, 'G', 5), (N, 'F', 8), (F, 'E', 8), (F, 'D', 12), (D, 'C', 7),
('C', 'B', 6), ('B', 'A', 6), ('C', 'O', 16), (M, 'P', 17), (O, 'Q', 18), (Q, 'S', 3), (
Q, 'R', 6)]

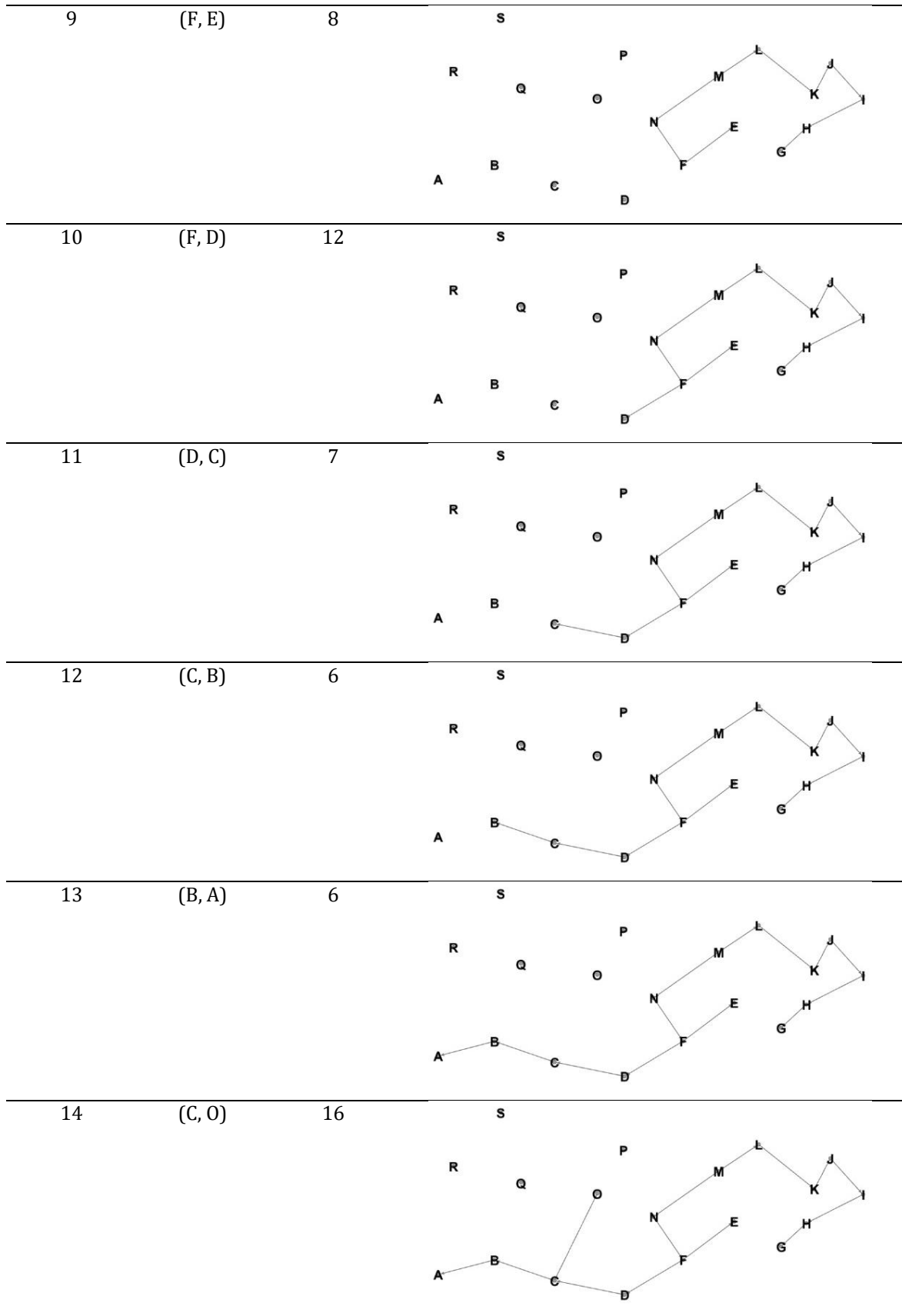
```

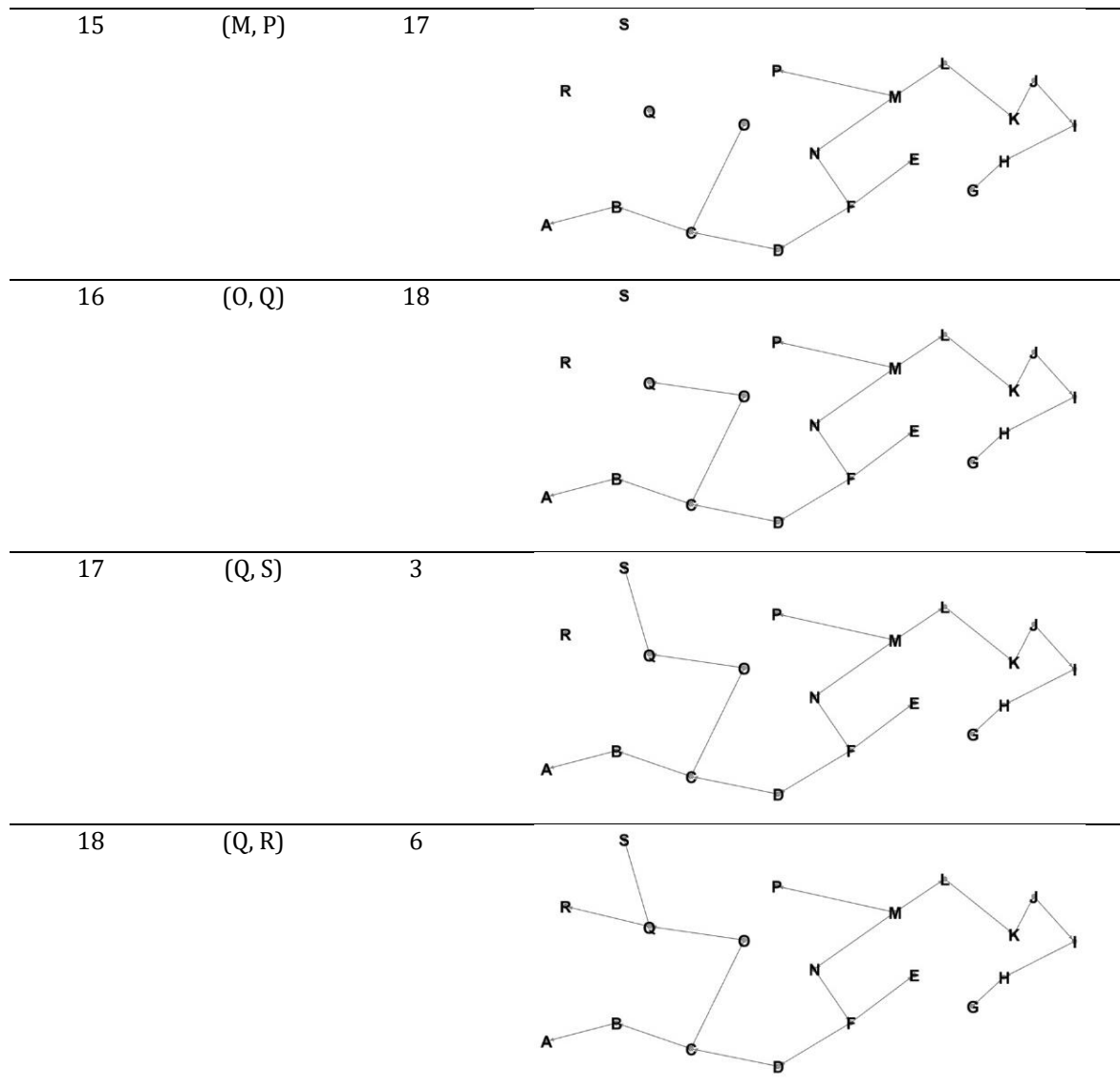
Selanjutnya hasil yang telah diperoleh akan divisualisasikan dengan graf sebagai berikut :

Tabel 2. *Minimum Spanning Tree* menggunakan Algoritma Prim

Langkah	Sisi	Bobot	Pohon Merentang Minimum
1	(M, N)	6	
2	(M, L)	8	







Untuk memberikan pemahaman lebih lanjut, berikut adalah penjelasan dari langkah-langkah yang telah divisualisasikan di atas :

Langkah 1

Diketahui M sebagai Ibukota Kabupaten Ngawi, yaitu Kecamatan Ngawi. Berdasarkan gambar graf di atas, dipilih sisi (M,N) dengan bobot 6 yang memiliki bobot minimal, kemudian digambarkan pada kolom *minimum spanning tree*.

Langkah 2

Pilih sisi yang berhubungan dengan simpul M atau N pada sisi (M, N) yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (M, L) dengan bobot 8 yang terhubung dengan simpul M pada sisi sebelumnya, yaitu (M, N), kemudian digambarkan pada *minimum spanning tree*.

Langkah 3

Pilih sisi yang berhubungan dengan simpul M, N, atau L yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (L, K) dengan bobot 7 yang terhubung dengan simpul L pada sisi (M, L), kemudian digambarkan pada *minimum spanning tree*.

Langkah 4

Pilih sisi yang berhubungan dengan simpul M, N, L, atau K yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (K, J) dengan bobot 6 yang terhubung dengan simpul K pada sisi (L, K), kemudian digambarkan pada *minimum spanning tree*.

Langkah 5

Pilih sisi yang berhubungan dengan simpul M, N, L, K, atau J yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (J, I) dengan bobot 3 yang terhubung dengan simpul J pada sisi (K, J), kemudian digambarkan pada *minimum spanning tree*.

Langkah 6

Pilih sisi yang terhubung dengan simpul I, J, K, M atau L yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (I, H) dengan bobot 4 yang terhubung dengan simpul I pada sisi (J, I), kemudian digambarkan pada *minimum spanning tree*.

Langkah 7

Pilih sisi yang terhubung dengan simpul H, I, J, K, L, atau M yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (H, G) dengan bobot 5 yang terhubung dengan simpul H pada sisi (I, H), kemudian digambarkan pada *minimum spanning tree*.

Langkah 8

Pilih sisi yang terhubung dengan simpul F, N atau M yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (N, F) dengan bobot 8 yang terhubung dengan simpul N pada sisi (M, N), kemudian digambarkan pada *minimum spanning tree*.

Langkah 9

Pilih sisi yang terhubung dengan simpul E, F, N, atau M yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (F, E) dengan bobot 8 yang terhubung dengan simpul F pada sisi (N, F), kemudian digambarkan pada *minimum spanning tree*.

Langkah 10

Pilih sisi yang terhubung dengan simpul D, F, N, atau M yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (F, D) dengan bobot 8 yang terhubung dengan simpul F pada sisi (N, F) dan (F, E), kemudian digambarkan pada *minimum spanning tree*.

Langkah 11

Pilih sisi yang terhubung dengan simpul D, F, N, M, atau L yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (D, C) dengan bobot 7 yang terhubung dengan simpul D pada sisi (F, D), kemudian digambarkan pada *minimum spanning tree*.

Langkah 12

Pilih sisi yang terhubung dengan simpul C, D, F, N, M, atau L yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (C, B) dengan bobot 6 yang terhubung dengan simpul C pada sisi (D, C), kemudian digambarkan pada *minimum spanning tree*.

Langkah 13

Pilih sisi yang terhubung dengan simpul B, C, D, F, N, M, atau L yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (B, A) dengan bobot 6 yang terhubung dengan simpul B pada sisi (C, B), kemudian digambarkan pada *minimum spanning tree*.

Langkah 14

Pilih sisi yang terhubung dengan simpul A, B, C, D, F, N, M, atau L yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (C, O) dengan bobot 16 yang terhubung dengan simpul C pada sisi (D, C), kemudian digambarkan pada *minimum spanning tree*.

Langkah 15

Pilih sisi yang terhubung dengan simpul A, B, C, D, F, M, N, L, atau O yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (M, P) dengan bobot 17 yang terhubung simpul M pada sisi (M, N) dan (M, L), kemudian digambarkan pada *minimum spanning tree*.

Langkah 16

Pilih sisi yang terhubung dengan simpul A, B, C, O, atau P yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (O, Q) dengan bobot 18 yang terhubung simpul O pada sisi (C, O), kemudian digambarkan pada *minimum spanning tree*.

Langkah 17

Pilih sisi yang terhubung dengan simpul A, B, P, atau Q yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (Q, S) dengan bobot 3 yang terhubung simpul Q pada sisi (O, Q), kemudian digambarkan pada *minimum spanning tree*.

Langkah 18

Pilih sisi yang terhubung dengan simpul A, B, Q, atau S yang memiliki bobot paling rendah dan tidak membentuk suatu *cycle*. Dipilih sisi (Q, R) dengan bobot 6 yang terhubung simpul Q pada sisi (O, Q), kemudian digambarkan pada *minimum spanning tree*.

Diperoleh hasil *Minimum Spanning Tree* menggunakan Visual Studio Code bahasa Python dengan jarak Kecamatan Sine – Kecamatan Ngrambe sejauh 6 km, Kecamatan Ngrambe – Kecamatan Jogorogo sejauh 6 km, Kecamatan Jogorogo – Kecamatan Kendal sejauh 7 km, Kecamatan Jogorogo – Kecamatan Kedunggalar sejauh 16 km, Kecamatan Kendal – Kecamatan Gerih sejauh 12 km, Kecamatan Geneng – Kecamatan Gerih sejauh 8 km, Kecamatan Gerih – Kecamatan Paron sejauh 8 km, Kecamatan Kwadungan – Kecamatan Pangkur sejauh 5 km, Kecamatan Pangkur – Kecamatan Karangjati sejauh 4 km, Kecamatan Karangjati – Kecamatan Bringin sejauh 3 km, Kecamatan Bringin – Kecamatan Padas sejauh 6 km, Kecamatan Padas – Kecamatan Kasreman sejauh 7 km, Kecamatan Kasreman – Kecamatan Ngawi sejauh 8 km, Kecamatan Ngawi – Kecamatan Paron sejauh 6 km, Kecamatan Ngawi – Kecamatan Pitu sejauh 17 km, Kecamatan Kedunggalar – Kecamatan Widodaren sejauh 18 km, Kecamatan Widodaren – Kecamatan Mantingan sejauh 6 km, Kecamatan Widodaren – Kecamatan Karanganyar sejauh 3 km.

Jadi, total keseluruhan jarak terpendek antar kecamatan di Kabupaten Ngawi tahun 2023 yang diperoleh dari Badan Pusat Statistik (BPS) dalam rangka pengoptimalan jaringan dengan pendekatan Minimum Spanning Tree (MST) menggunakan Algoritma Prim adalah sepanjang 146 km.

PENUTUP

Dengan memanfaatkan Algoritma Prim diperoleh *Minimum Spanning Tree* sehingga diketahui jalur terpendek antar 19 kecamatan di Kabupaten Ngawi sejauh 146 km. Dan hasil tersebut dapat dimanfaatkan untuk optimalisasi jaringan jalan yang dapat mendukung pemerataan pembangunan, optimalisasi jaringan transportasi, kemudahan akses, bahkan dasar bagi sistem navigasi dan logistik di dalam kota.

DAFTAR PUSTAKA

- [1] Minarwati. (2021). Aplikasi Minimum Spanning Tree Algoritma Prim Dan Kruskal Penentuan Pembangunan jalan Baru, *Fahma: Jurnal Informatika komputer, Bisnis, dan Manajemen*, 19(2), 51–60.
- [2] Buhaerah, Busrah, Z., & Sanjaya, H. (2019). *Teori Graf dan Aplikasinya*. In *Living Spiritual Quotient*.
- [3] Nugraha, D. W. (2011). Aplikasi Algoritma Prim untuk Menentukan Minimum Spanning Tree Suatu Graf Berbobot Berorientasi Objek. *Teknik Elektro UNTAD Palu*, 1(2), 70–79.
- [4] Sembiring, R. R., Sufri, & Multahadah, C. (2022). Penerapan Algoritma Prim dalam Menentukan Minimum Spanning Tree (MST) (Studi Kasus: Jaringan Pipa PDAM Tirta Muaro Jambi). *Jurnal Ilmiah Matematika Dan Terapan*, 19(1), 58–71. <https://doi.org/10.22487/2540766x.2022.v19.i1.15890>
- [5] Suhika, D., Muliawati, T., & Ruwandar, H. (2020). Optimalisasi Rencana Pemasangan Kabel Fiber Optic Di Itera Dengan Algoritma Prim. *AKSIOMA: Jurnal Program Studi Pendidikan Matematika*, 9(1), 86. <https://doi.org/10.24127/ajpm.v9i1.2597>
- [6] Fitriya B, W. A., Rosnafi'an Sumardi, S., Paranoan, N. R., Bintang, C., & Allo, G. (2023). Penentuan Rute Di Aplikasi Google Maps Dengan Menggunakan Graf Dan Algoritma Prim. *Jurnal Multidisiplin Ilmu*, 2(1), 2828–6863.
- [7] Rahmadi, D., & Sandariria, H. (2023). Penerapan Minimum Spanning Tree dalam Menentukan Rute Terpendek Distribusi Naskah Soal USBN di SMA Negeri se-Sleman. *Basis: Jurnal Ilmiah Matematika*, 2(1), 66–71.
- [8] Rahmadi, D., Maharani, N. P., Syifa, M. R., Sama, S. A., and Ardiansyah, G. F., Penerapan Minimum Spanning Tree dalam Menentukan Rute Terpendek Distribusi Naskah Soal USBN di SMA Negeri se-Sleman, *Journal of Mathematics Theory and Applications*, 2(2) (2023), 24-33.
- [9] Rahmadi, D., & Herdianti, R. (2024). Penerapan Minimum Spanning Tree dalam Menentukan Rute Terpendek pada Wisata di Kota Wonogiri. *Basis : Jurnal Ilmiah Matematika*, 3(2), 31-39. doi:10.30872/basis.v3i2.1390
- [10] Albar, W., Rahmadi, D., & Dewi, K. (2023). The Implementation of Minimum Spanning Tree in Finding Algebraically the Shortest Path of National-Exam-Sheet Distribution in All Senior High Schools over Bantul Regency. *Basis : Jurnal Ilmiah Matematika*, 2(1), 78-82. doi:10.30872/basis.v2i1.1111