

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Studi Pustaka**

Perkembangan ekonomi suatu wilayah sangat dipengaruhi oleh pariwisata. Kelangkaan, kealamian, orisinalitas, pemberdayaan masyarakat, optimalisasi lahan, pemerataan, dan transportasi merupakan aspek pendukung pariwisata. Salah satu variabel yang dapat dioptimalkan dalam pariwisata adalah transportasi. Membangun jaringan transportasi wisata yang efektif dan efisien merupakan strategi terbaik untuk memaksimalkan unsur transportasi. Salah satu unsur yang menunjang keberhasilan industri pariwisata suatu daerah adalah memiliki infrastruktur transportasi yang memadai. Promotor pengembangan pariwisata dapat berupa fasilitas transportasi umum khusus. Wisatawan yang ingin mengunjungi tempat wisata dapat melakukan perjalanan dengan cepat, mudah, dan efektif menggunakan fasilitas transit tersebut (Rifanti & Arifwidodo, 2019).

Pengunjung membutuhkan informasi mengenai jalur wisata untuk membantu mereka merencanakan perjalanan saat menuju destinasi, yaitu untuk kembali ke rumah asal atau lokasi sementara. Demi menghemat waktu, uang, dan jarak tempuh, wisatawan kerap mencari jalur terpendek menuju destinasi wisata (Widya & Andrasto, 2019). Pada umumnya, masyarakat yang ingin melakukan perjalanan wisata tentunya akan membuat rencana terlebih dahulu mengenai lokasi yang akan dikunjungi beserta waktu keberangkatannya. Hal ini dilakukan untuk menghindari timbulnya permasalahan salah satunya yaitu jarak tempuh yang akan dilalui ternyata lebih jauh sehingga waktu yang dibutuhkan tidak sesuai dengan harapan. Oleh karena itu diperlukan petunjuk untuk mendampingi wisatawan yang berkunjung ke tempat-tempat wisata di Kabupaten Banyumas, Jawa Tengah (Sologia dkk., 2020).

Saat menentukan rekomendasi rute wisata dapat menggunakan metode *Travelling Salesman Problem (TSP)*. Metode *Travelling Salesman Problem (TSP)*, merupakan pencarian rute terpendek dengan seorang *salesman* mengunjungi setiap *node* tepat hanya sekali dan akan kembali ke titik awal keberangkatan. Pada kasus

ini, pemilihan metode TSP dapat dianalogikan *salesman* merupakan wisatawan yang akan mengunjungi objek wisata (*node*) yang telah ditentukan sebelumnya oleh wisatawan tepat satu kali dengan titik awal dan akhir yaitu alun-alun, terminal dan stasiun Purwokerto (Sologia dkk., 2020). Salah satu metode yang digunakan pada penelitian ini yaitu metode *Travelling Salesman Problem* (TSP), dengan menggunakan model *Pemrograman Linier Integer* atau *Integer Linier Programming* (ILP) yang dapat membantu dalam menyelesaikan masalah TSP. *Integer Linier Programming* (ILP) adalah masalah optimasi dengan fungsi objektif dan kendala yang linier serta *variable integer* (Paillin & Tupan, 2021).

Berbagai penelitian telah dilakukan untuk mengkaji masalah yang berkaitan dengan pencarian rute terpendek. Beberapa di antaranya mengkaji pencarian rute terpendek dalam bidang pariwisata dan transportasi menggunakan *Travelling Salesman Problem* (TSP) seperti penelitian yang dilakukan pada PT. Paris Jaya Mandiri dengan metode *Integer Linier Programming* (ILP) dalam menyelesaikan masalah rute optimal untuk melakukan distribusi produk Nestle dari gudang menuju *outlet* dengan hasil efisiensi rute tempuh baik dari segi jarak sebesar 26.32% dan waktu total 7.42% (Paillin & Tupan, 2021). Selanjutnya penelitian menggunakan metode TSP juga dilakukan untuk mencari rekomendasi rute wisata di Toraja Utara agar dapat menemukan rute yang optimal dengan mengimplementasikan algoritme *K-Nearest Neighbor*. Hasil dari penelitian ini yaitu pada hari ke-1 terjadi penghematan jarak 7,08 km, hari ke-2 penghematan jarak 14,28 km dan pada hari ke-3 penghematan jarak sebesar 50,94 km (Sologia dkk., 2020).

Penelitian lainnya terkait penggunaan metode TSP juga pernah dilakukan untuk mengetahui jalur terbaik yang menghubungkan 11 objek wisata di Banyumas menggunakan Algoritme Genetika dengan hasil panjang jalur terbaik yaitu 0.878 unit kartesian dengan ukuran populasi 25 dan probabilitas mutasi 0.005. Saran dari penelitian ini yaitu perlunya perbandingan hasil yang diperoleh dengan metode yang lain seperti linear *programming* atau jaringan saraf tiruan (Tahyudin & Susanti, 2015).

Selain itu, terdapat penelitian yang menggunakan algoritma *Dijkstra* untuk melihat rute terpendek dalam industri perjalanan dan pariwisata, khususnya

penelitian yang dilakukan untuk menemukan jalur terbaik untuk mengurangi lalu lintas di Purwokerto. Untuk menentukan jalur optimal yang dapat digunakan penduduk di Purwokerto, penelitian ini mengkaji sejumlah simpang Purwokerto yang rawan kemacetan lalu lintas. Namun penelitiannya hanya melihat jalur transportasi untuk mengurangi kemacetan lalu lintas, bukan jalur wisata di Purwokerto. (Rifanti, 2017).

Penelitian mengenai rute terpendek pada transportasi pariwisata juga dapat menggunakan teori graf, salah satunya yaitu algoritme *Floyd* seperti yang dilakukan dalam merancang rute bus wisata di Banyumas untuk mengetahui rute terpendek. Temuan investigasi ini mengungkap rute yang bisa ditempuh bus wisata untuk menempuh jarak terendah. Penataan jalur transit ini diharapkan dapat mempromosikan pariwisata Kabupaten Banyumas baik dari wisatawan domestik maupun mancanegara (Rifanti & Arifwidodo, 2019).

Berdasarkan penelitian terdahulu yang telah dijelaskan di atas dapat menjadi bahan referensi dalam penelitian untuk mencari rute terpendek pada objek wisata di Kabupaten Banyumas. Rincian dari penelitian sejenis tersebut dapat dijelaskan melalui Tabel 2.1.

Tabel 2.1 Perbandingan Metode dan Objek Penelitian Sebelumnya

No	Judul (Tahun)	Penulis	Objek	Metode
1	Implementasi Algoritme Dijkstra dalam Pencarian Rute Terpendek Tempat Wisata di Kabupaten Klaten (2020)	Nugroho Arif Sudibyo, Permadi Eka Setyawan, Yohana Putra Surya Rahmad Hidayat	Wisata Air di Kabupaten Klaten.	Algoritme Dijkstra
2	Penentuan Rute Terpendek Tujuan Wisata di Kota Toboali Menggunakan	Fransiskus Panca Juniawan, Dwi Yuny Syifania	Wisata di Kota Toboali	Algoritme Dijkstra Berbasis Web

No	Judul (Tahun)	Penulis	Objek	Metode
	Algoritme Dijkstra Berbasis Web (2019)			
3	Pemanfaatan Metode <i>Hill Climbing</i> Mencari Jalur Terpendek Objek Wisata Kabupaten Lima Puluh Kota (2022)	Dian Permata Sari	Objek Wisata Kabupaten Lima Puluh Kota	<i>Hill Climbing</i>
4	Implementasi Algoritme <i>Floyd</i> dalam Menentukan Rute Terpendek Transportasi Pariwisata (2019)	Utti Marina Rifanti, Bongga Arifwidodo	Objek wisata di Kabupaten Banyumas	Algoritme <i>Floyd</i>
5	Analisis Rute Pengangkutan Sampah Kota Manado dengan Pendekatan <i>Vehicle Routing Problem</i> (VRP) (2019)	Andre Ch. Lasut, Friska M. Makalew, Prudensy F. Opit	Pendistribusian Sampah Daerah Aliran Sungai (DAS) di Kota Manado	<i>Vehicle Routing Problem</i> (VRP)
6	Aplikasi Model VRP dan Metode <i>Saving Matrix</i> untuk Mengoptimalkan Rute Pendistribusian Pupuk di CV. Al-Zaman (2021)	Koko Hermanto, Silvia Firda Utami, Wari Ammar Abdul Jabbar	CV. Al-Zaman	Model VRP dan Metode <i>Saving Matrix</i>

No	Judul (Tahun)	Penulis	Objek	Metode
7	Model <i>Integer Linear Programming</i> (ILP) dalam Pemecahan <i>Traveling Salesman Problem</i> (TSP) (2020)	Daniel B. Paillin, Johan M Tupan	PT. Paris Jaya Mandiri-Ambon	Model <i>Integer Linear Programming</i> (ILP) dan <i>Traveling Salesman Problem</i> (TSP)
8	Optimasi Pendistribusian Produk Menggunakan Metode <i>Integer Linear Programming</i>	Safrudin Ismail, Ismail Djakaria, Djihad Wungguli	PT. Awet Sarana Sukses Gorontalo	<i>Integer Linear Programming</i>
9	Rekomendasi Rute Wisata Menggunakan Metode <i>Travelling Salesman Problem</i> dengan Algoritme <i>K-Nearest Neighbor</i>	Friska Sologia, Rio Aurachman, Putu Giri Artha Kusuma	Objek wisata di Toraja Utara	<i>Travelling Salesman Problem</i> dengan Algoritme <i>K-Nearest Neighbor</i>
10	<i>Application Of Linier Fuzzy Multi-Objective Programming in Travelling Salesman Problem</i> (2021)	E Hertini, J Nahar, A K Supriatna	PT. X (Sei Raya, Kota Baru, Siantan, Adisucipto, Gajahmada)	<i>Linier Fuzzy Multi-Objective Programming, TSP</i>

Berdasarkan rincian pada Tabel 2.1 dapat diketahui bahwa penelitian pencarian rute terpendek objek khusus wisata di Kabupaten Banyumas belum ada yang melakukan penelitian. Penelitian yang telah dilakukan sebelumnya mengenai rute terpendek di kabupaten Banyumas menggunakan metode TSP (Tahyudin &

Susanti, 2015). Selain itu, penelitian sebelumnya hanya meneliti jalur terpendek khusus untuk rute bus pariwisata di kabupaten Banyumas (Rifanti & Arifwidodo, 2019). Metode yang banyak digunakan dalam mencari rute terpendek yaitu TSP, *Floyd* dan *Dijkstra*. Oleh sebab itu, penelitian ini akan menggunakan metode *Travelling Salesman Problem* (TSP) dengan metode *linear programming* agar dapat menemukan rute terpendek untuk objek wisata di kabupaten Banyumas.

## **2.2 Dasar Teori**

### **2.2.1 Pariwisata di Kabupaten Banyumas**

Kabupaten Banyumas menawarkan berbagai tempat wisata, baik wisata alam, religi, budaya, maupun edukasi, termasuk museum. Pemerintah daerah sedang berupaya untuk meningkatkan kawasan wisata ini, yang berujung pada peningkatan jumlah objek wisata yang terdaftar di BPS Kabupaten Banyumas, dari 10 pada tahun 2002 menjadi 17 pada tahun 2021. Wisata alam, wisata budaya, dan wisata religi merupakan tiga kategori wisata yang populer di wilayah Kabupaten Banyumas dengan pengunjung baik domestik maupun mancanegara. Dari 11 tempat wisata di tahun 2009 menjadi 13 tempat wisata di tahun 2011, perkembangannya sangat pesat. Pada tahun itu, Desa Pancasan, wilayah administrasi Kabupaten Banyumas Kecamatan Ajibarang mengizinkan tumbuhnya wisata alam, khususnya wisata air (Hermawati & Milawaty, 2016). Berdasarkan pada data Dinas Pemuda, Olahraga, Kebudayaan dan Pariwisata Provinsi Jawa Tengah, jumlah daya tarik wisata Kabupaten Banyumas pada tahun 2021 mencapai 93 dengan rincian 42 wisata alam, 4 wisata budaya, 34 wisata buatan, 3 wisata minat khusus, dan 10 wisata lain-lain. Kabupaten Banyumas juga memperoleh peringkat 3 di Jawa Tengah dengan pengunjung wisatawan nusantara terbanyak dengan jumlah 1.498.236 orang.

### **2.2.2 Linear Proramming**

Solusi terbaik yang bisa dibayangkan untuk masalah pembagian sumber daya yang langka di antara aktivitas yang bersaing adalah dengan menggunakan *linear programming*. Ketika seseorang harus memilih tingkat aktivitas tertentu yang bersaing untuk sumber daya terbatas yang diperlukan untuk melaksanakan aktivitas tersebut, masalah alokasi sumber daya akan terjadi. Jadi, pemrograman linier adalah

teknik untuk mendapatkan hasil sebesar mungkin untuk memenuhi tujuan terbaik di antara semua opsi. Dengan sumber daya terbatas yang dimiliki oleh perusahaan atau organisasi dan diorganisasikan ke dalam fungsi kendala yang juga linier di antara berbagai penggunaan yang bersaing, para ahli mengklaim bahwa pemrograman linier adalah teknik matematika yang digunakan untuk menemukan jawaban optimal untuk fungsi tujuan linier. (Paillin & Tupan, 2021).

Fungsi tujuan *linier* ditambah persamaan linier dan *non-linier* yang membentuk variabel kendala menjadikan pemrograman *linier* sebagai masalah optimisasi. Setiap kendala dapat mengambil bentuk yang berbeda tergantung pada masalahnya. (Sulistiyono, 2022). Pemrograman bilangan bulat biner adalah varian lain dari masalah pemrograman bilangan bulat (BIP). Dalam masalah pemrograman bilangan bulat biner, bilangan biner berfungsi sebagai nilai variabel keputusan (0 atau 1). Dalam aplikasi dunia nyata, pemrograman bilangan bulat biner berurusan dengan dilema pengambilan keputusan, di mana jawabannya adalah angka 1 yang menunjukkan "ya" atau angka 0 yang menunjukkan "tidak".(Paillin & Tupan, 2021).

Definisi 3 Masalah *Linear Programming* (LP) merupakan masalah optimisasi dengan ketentuan sebagai berikut:

1. Untuk memaksimumkan atau meminimalkan fungsi linear dari variabel keputusan. Fungsi yang akan dimaksimumkan atau diminimalkan disebut fungsi objektif.
2. Nilai dari variabel keputusan harus memenuhi semua fungsi kendala. Masing-masing fungsi kendala harus berupa persamaan linear atau pertidaksamaan linear
3. Fungsi pembatas berkaitan dengan masing-masing variabel. Untuk setiap variabel  $x_i$ , fungsi pembatas menetapkan bahwa  $x_i$  harus bernilai non-negatif ( $x_i \geq 0$ ) atau tidak dibatasi dalam fungsi (Ismail dkk., 2021).

### 2.2.3 *Travelling Salesman Problem (TSP)*

*Travelling salesman problem (TSP)* muncul ketika seorang salesman melakukan perjalanan ke beberapa lokasi yang telah ditentukan dan mengetahui jarak antara setiap lokasi. *Travelling salesman problem (TSP)* merupakan tantangan

bagi tenaga penjual yang harus meninggalkan suatu depo dan pergi ke  $n$  node atau kota, dengan batasan setiap node hanya dapat dikunjungi satu kali, sebelum kembali ke depot keberangkatan melalui jalur terpendek. Rute yang menawarkan biaya keseluruhan, waktu perjalanan, dan jarak terendah adalah pilihan terbaik. Hasil dari routing adalah daftar tempat-tempat yang harus dikunjungi oleh tenaga penjual selama satu kali tour (Ardana & Saputra, 2016).

Berikut adalah aturan-aturan yang mengidentifikasi bahwa permasalahan tersebut adalah TSP:

1. Perjalanan dimulai dan diakhiri di kota yang sama dengan tempat dimulainya penjualan.
2. Tanpa melewati satu kota pun, semua kota harus dikunjungi.
3. Penjual tidak dapat kembali ke kota awal mereka sebelum melakukan tur ke setiap kota.
4. Dengan menyesuaikan urutan kota, dimungkinkan untuk mendapatkan nilai optimal sekaligus mengurangi jarak keseluruhan yang ditempuh (Yulianto & Setiawan, 2018)

Parameter:

$N$  = Jumlah kota/lokasi/pelanggan yang akan dikunjungi. Perhatikan bahwa itu tidak termasuk basis, yang di indeks oleh  $i = 0$

$C_{ij}$  = Biaya/jarak perjalanan dari kota  $i$  ke kota  $j$

$A$  = Himpunan busur  $(i,j)$  yang ada. Perhatikan bahwa dengan  $(i, j)$  yang kami maksud hanya busur yang ada dari *node*  $i$  ke *node*  $j$

Variabel:

$X_{ij} = \begin{cases} 1 & \text{jika wisatawan melakukan perjalanan dari kota } i \text{ ke kota } j \\ 0 & \text{sebaliknya} \end{cases}$

TSP model

Minimasi

$$Z = \sum_{(i,j) \in A} c_{ij} x_{ij} \dots\dots\dots (1)$$

Kendala

1. *Salesman* harus mengunjungi semua lokasi tepat satu kali

$$\sum_{\substack{i=0 \\ i \neq j}}^n x_{ij} = 1, \forall j = 0, \dots, n \dots\dots\dots (2)$$



$$\sum_{\substack{j=0 \\ i \neq j}}^n x_{ij} = 1, \forall i = 0, \dots, n \dots\dots\dots (3)$$

2. Tidak boleh ada *subtour* yang terbentuk dalam perjalanan pada TSP, sehingga hanya akan membentuk satu jalur tur dari TSP

$$u_i - u_j + nx_{ij} \leq n - 1, i \neq j; \forall i = 2, 3, \dots, n; \forall j = 2, 3, \dots, n; u_j \geq 0) \dots\dots\dots (4)$$

Traveling Salesman Problem (TSP), seperti yang didefinisikan oleh Johnson dan McGeoch (1997), adalah tugas menentukan rute tertutup untuk mengunjungi sejumlah kota, di mana setiap kota hanya dikunjungi sekali dan setelah semua pelancong kembali ke kota pertama. kota telah dikunjungi. Banyak metode seperti *algoritme genetic*, *algoritme simulated annealing*, *algoritme tabu search*, *algoritme lin Kernighan*, *algoritme neural network*, dan *algoritme local search* telah digunakan untuk memecahkan masalah TSP, menurut Johnson dan McGeoch (1997). (Sarker & Newton, 2008).

#### 2.2.4 Aplikasi LINGO

LINGO adalah alat menyeluruh yang dibuat untuk menyelesaikan model pengoptimalan linier, nonlinier, kuadrat, dibatasi kuadrat, kerucut orde kedua, stokastik, dan bilangan bulat dengan lebih cepat, sederhana, dan efektif (Kello dkk.,2017). LINDO (2011) menegaskan bahwa model LINGO harus mempertimbangkan faktor-faktor berikut:

- a. Gaya ini mengharuskan komentar dimulai dengan tanda seru (!) dan muncul dalam teks berwarna hijau.
- b. Teks biru menunjukkan fungsi operator yang telah ditentukan di LINGO.
- c. Teks hitam akan digunakan untuk teks lain. D. Dalam LINGO, setiap kalimat harus diakhiri dengan (;)
- d. Nama variabel harus berdasarkan abjad (A–Z). Huruf, kata, dan karakter garis bawah ( \_ ) adalah karakter tambahan yang dapat diterima untuk nama variabel. Panjang nama variabel maksimal 32 karakter.

Fitur yang dapat digunakan untuk menyelesaikan permasalahan optimasi pada Lingo 11 sebagai berikut:

- a. *Command SETS*

Dalam aplikasi LINGO, perintah SETS digunakan untuk mengelompokkan elemen dari variabel yang sama. SETS digunakan pada program LINGO sebelum model *constraint* dan diakhiri dengan perintah ENDSETS. Aplikasi LINGO juga dapat menggunakan berbagai rangkaian fungsi. Berikut adalah fungsi-fungsinya (LINDO, 2020):

1) *@FOR*

Digunakan dalam batasan untuk menentukan setiap *constraint*.

2) *@SUM*

Mendeklarasikan jumlah pernyataan di semua *member set*.

3) *@MIN*

Digunakan untuk menghitung nilai minimum pernyataan dari semua *member set*.

4) *@MAX*

Fungsi ini digunakan untuk menghitung nilai maksimum suatu pernyataan dari semua *member set*.

b. LINGO DATA *Section*

Untuk menetapkan setiap nilai dari variabel yang berbeda menggunakan *command DATA*, LINGO 11 menyediakan bagian yang terpisah. Di setiap model Lingo 11, perintah DATA ditulis setelah perintah SETS. Data dan EndData masing-masing adalah label untuk bagian ini. Frasa "*object\_list = value\_list*" mengikuti deklarasi DATA. Nama setiap atribut dari suatu himpunan yang telah diberi nilai dicantumkan dalam *Object list*.

c. Operasional dan Fungsi LINGO 11

LINGO 11 menyediakan fungsi dan operasional dalam model sebagai pemecahan masalah. LINGO 11 mempekerjakan tiga macam operasi yaitu operasi aritmatika, logika, dan relasional. Berikut ini adalah operasi aritmatika:

1) Eksponen ( ^ )

2) Perkalian ( \* )

3) Pembagian ( / )

4) Penjumlahan ( + )

### 5) Pengurangan ( - )

Dalam set fungsi, operasi logis digunakan untuk menentukan kondisi *TRUE/FALSE* menurut LINDO (2011):

- 1) *#LT#* = *TRUE* ketika nilai di bagian kiri secara tepat kurang dari nilai bagian kanan, lainnya *FALSE*
- 2) *#LE#* = *TRUE* ketika nilai di bagian kiri kurang dari atau sama dengan nilai di bagian kanan, lainnya *FALSE*
- 3) *#GT#* = *TRUE* ketika nilai di bagian kiri secara tepat lebih besar dari nilai di bagian kanan, lainnya *FALSE*
- 4) *#GE#* = *TRUE* ketika nilai di bagian kiri secara lebih besar atau sama dengan nilai di bagian kanan, lainnya *FALSE*
- 5) *#EQ#* = *TRUE* jika kedua nilai sama, lainnya *FALSE*
- 6) *#NE#* = *TRUE* jika kedua nilai tidak sama, lainnya *FALSE*
- 7) *#AND#* = *TRUE* jika kedua nilai bernilai *TRUE*, lainnya *FALSE*
- 8) *#OR#* = *FALSE* hanya jika kedua nilai bernilai *FALSE*, lainnya *TRUE*
- 9) *#NOT#* = *TRUE* ketika nilai adalah *FALSE*, lainnya *FALSE*

Dalam sebuah model diperlukan relasi operasional untuk mendefinisikan batasan-batasan, antara lain:

- a. Ekspresi sama dengan (=)
- b. Pernyataan di bagian kiri kurang dari atau sama dengan bagian kanan ( $\leq$ )
- c. Pernyataan di bagian kiri lebih besar dari atau sama dengan bagian kanan ( $\geq$ ) (LINDO, 2020)

#### 2.2.5 Aplikasi *Google Maps*

Layanan peta elektronik gratis yang ditawarkan oleh *Google* disebut *Google Maps*. Untuk dapat menggunakan *Google Maps* dengan mengunjungi URL <http://maps.google.com>. *Google Maps* untuk menghitung jarak antara dua lokasi. Untuk mengetahui jarak yang harus ditempuh dapat dilakukan dengan memasukkan nama tempat, koordinat, atau alamat lokasi yang posisi dan jaraknya dari titik awal atau saat ini yang ingin diketahui. Langkah pertama yang harus dilakukan adalah memasukkan informasi lokasi suatu tempat di kolom pencarian untuk

mengidentifikasi tempat yang ingin diketahui jarak dan lokasi daerah tersebut. Dalam hal ini, bujur dan lintang dari dua titik digunakan sebagai data lokasi untuk menghitung jarak. (Ardana & Saputra, 2016).