II. TINJAUAN PUSTAKA

2.1 Graf

Menurut Aldous dan Wilson (2000), graf merupakan diagram yang mempresentasikan titik yang disebut simpul sebagai objek dan garis yang disebut sisi mempresentasikan hubungan antar objek. Secara sederhana graf didefinisikan sebagai kumpulan titik yang dihubungkan oleh garis atau sisi.

Secara matematis, graf didefinisikan sebagai berikut :

Definisi menurut Rinaldi Munir (2009), graf G didefinisikan sebagai pasangan himpunan (V,E), ditulis dengan notasi G = (V, E), yang dalam hal ini :

V = himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*)

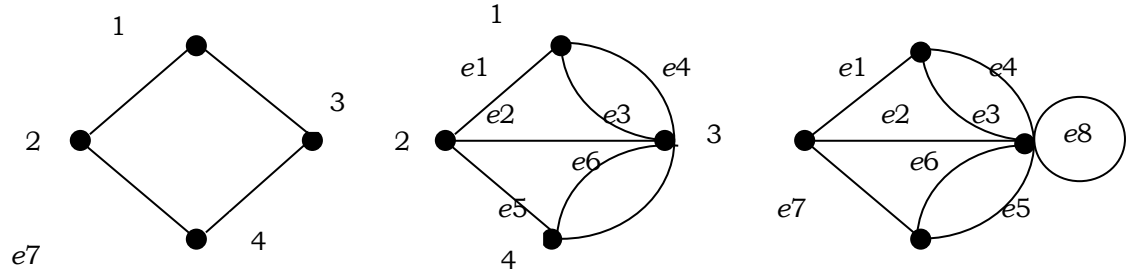
= $\{v_1, v_2, \ldots, v_n\}$

E = himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul

= $\{e_1, e_2, \ldots, e_n\}$

Definisi ini menyatakan bahwa V tidak boleh kosong, sedangkan E boleh kosong. Jadi, sebuah graf dimungkinkan tidak mempunyai sisi satu buah pun, tetapi simpulnya harus ada, minimal satu. Graf yang hanya mempunyai satu buah simpul tanpa sebuah sisi dinamakan graf trivial.

Simpul pada graf dapat diberi nomor dengan huruf, seperti a, b, c, ..., v, w, dengan bilangan asli 1, 2, 3, ... atau gabungan keduanya. Sedangkan sisi yang menghubungkan simpul dinyatakan dengan lambang *e*1, *e*2, ... , *en*. Secara geometri graf digambarkan sebagai sekumpulan noktah (simpul) di dalam bidang dua dimensi yang dihubungkan dengan sekumpul garis (sisi).



G1 G2 G3

Gambar 1. Contoh Graf Secara Umum

Sumber: Rinaldi Munir,Buku Matematika Diskrit Edisi Revisi Kelima 2012

Pada gambar 1 Ditunjukkan G1 graf sederhana yaitu graf yang tidak memiliki sisi ganda atau gelang (*loop*) ,G2 merupakan Graf ganda yang terdapat sisi ganda, dan pada G3 merupakan Graf semu yang dilengkapi dengan gelang (*loop*).

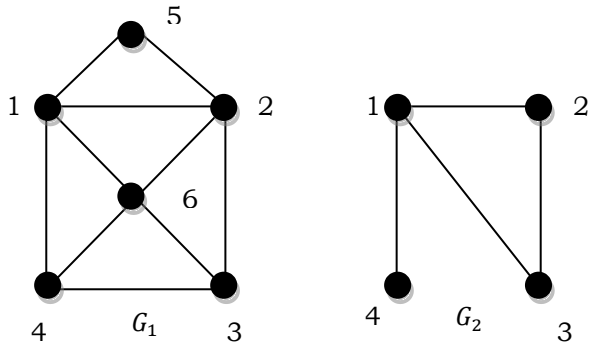
2.2 Struktur Graf

Menurut Heri Purwanto, dkk (2006), ada beberapa struktur dalam sebuah graf yaitu sebagai berikut :

1. Simpul merupakan himpunan node atau titik pada suatu graf.
2. Sisi merupakan himpunan garis yang mengaitkan tiap node atau titik.
3. Lintasan merupakan serangkaian simpul-simpul yang berbeda, yang secara berturut-turut dari simpul satu kesimpul selanjutnya. Lintasan dibagi menjadi 2 yaitu lintasan Euler dan lintasan Hamilton.

a. Lintasan Euler yaitu lintasan yang melewati masing-masing sisi dalam graf tepat satu kali.
b. Lintasan Hamilton yaitu lintasan yang melewati setiap simpul didalam graf tepat satu kali.

Graf yang memiliki lintasan Euler disebu graf semi Euler dan graf yang memiliki lintasan Hamilton disebut dengan graf semi Hamilton. (Sutarno, dkk., 2003)



Gambar 2. Graf Semi Euler dan Gral Semi Hamilton

Sumber: Rinaldi Munir, Buku Matematika Diskrit ldisi Revisi Kelima 2012

LintasanG1=4—1—5—2—1—6—2—3—4—6—3

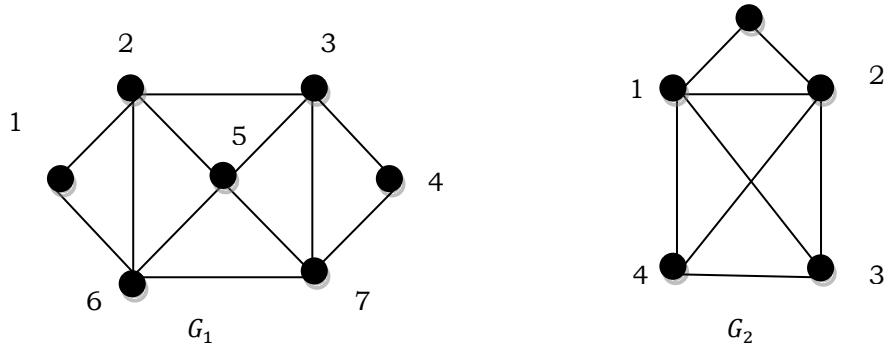
LïntasanG2=4—1—2—3

Jadi G1 merupakan graf semi Euler dan G merupakan graf semi Hamilton.

4. Sirkuit - Sirkuit atau siklus merupakan lintasan yang berawal dan berakhir pada simpul yang sama. Sirkuit juga ada 2 yaitu sirkuit Euler dan sirkuit Hamilton.

a. Sirkuit Euler yaitu sirkuit yang melewati masing-masing sisi di dalam graf tepat satu kali.
b. Sirkuit Hamilton yaitu sirkuit yang melewati tiap simpul di dalam graf tepat satu kali.

Graf yang mempunyai sirkuit Euler disebut juga graf Euler dan graf yang mempunyai sirkuit Hamilton disebut graf Hamilton. (Munir, 2012)



Gambar 3. Graf Euler dan Graf Hamilton

Sirkuit G1 = 2-3-4-7-3-5-7-6-2-5-6-1-2

Sirkuit G2 = 1-5-2-3-4-1

Jadi G1 merupakan graf Euler dan G2 merupakan graf Hamilton.

2.3 Terminologi graf

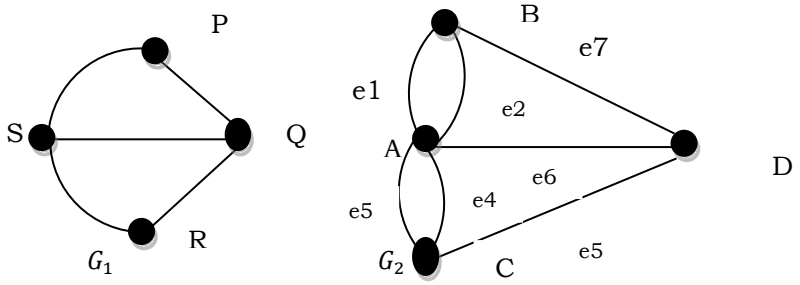
Menurut Rinaldi Munir (2012), terdapat terminologi-terminologi pada graf yang perlu diketahui, yaitu:

1. Ketetanggaan

Dua buah simpul dikatakan bertetangga jika kedua simpul tersebut terhubung langsung oleh suatu sisi. Pada graf ini, simpul P bertetangga dengan simpul Q dan S, tetapisimpul P tidakbertetanggadengansimpul R.

2. Bersisian

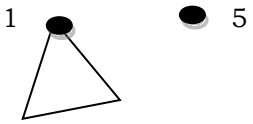
Suatu sisi e dikatakan bersisian dengan simpul v_1 dan simpul v_2 jika e menghubungkan kedua simpul tersebut, dengan kata lain e=(v_1, v_2). Contohnya graf dari masalah jembatan Konigsberg, maka e bersisian dengan simpul A dan simpul C, tetapi sisi tidak bersisian dengan simpul B. Graf Ketetanggaan dan Bersisian dapat dilihat pada Gambar 4.

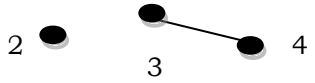


Gambar 4. Graf Ketetanggan dan Bersisian

3. Simpul terisolasi

Simpul dapat dikatakan simpul terisolasi jika simpul tersebut tidak memiliki sisi yang bersisian dengannya. Atau, dapat dinyatakan bahwa simpul yang tidak memiliki tetangga, seperti pada Gambar 5 simpul terisolasi ditunjukkan pada simpul 5.





Gambar 5. Graf dengan simpul terisolasi

4. Graf kosong

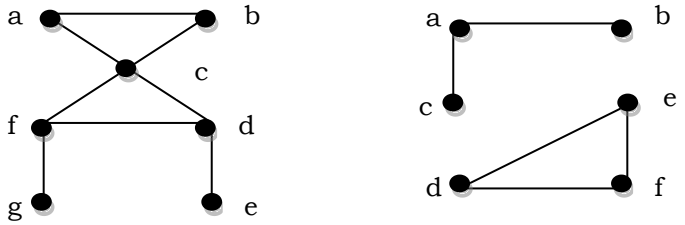
Graf kosong merupakan graf yang himpunan sisinya adalah himpunan kosong. Penulisan graf kosong dapat berupa N dengan n merupakan jumlah simpul.

5. Derajat

Derajat suatu simpul pada graf tidak berarah adalah jumlah sisi yang bersisian dengan simpul tersebut. Dengan notasi $d(v)$ yang menyatakan derajat dari simpul v. Pada graf berarah derajat dibedakan menjadi dua yaitu derajat masuk dan derajat keluar. Derajat masuk adalah jumlah sisi yang masuk pada simpul tersebut. Sedangkan derajat keluar adalah jumlah sisi yang keluar dari simpul tersebut.

6. Terhubung

Suatu graf tidak berarah dapat dikatakan graf terhubung jika untuk setiap pasang simpul yang ada pada G1 dan G2 terdapat lintasan. Jika terdapat saat saja pasangan yang tidak memiliki lintasan, maka graf tersebut dikatakan graf tidak terhubung. Perbedaan graf terhubung dan tak terhubung dapat dilihat pada Gambar 6.



Gambar 6. Graf terhubung dan graf tidak terhubung

Sumber: Kenneth H. Rosen – Discrete Mathematics and its Application, 7^{th}

2.4 *Travelling Salesman Problem* (TSP)

Travelling Salesman Problem (TSP) dikemukakan pada tahun 1800 oleh matematikawan Irlandia, Willian Rowan Hamilton dan matematikawan inggris Thomas Penyngton. *Travelling Salesman Problem* (TSP) sangat banyak aplikasinya dalam dunia nyata dan banyak menarik perhatian para peneliti sejak beberapa dekade terdahulu (Eistele, Castaneda, dan Galindo, 2002;

Reinelt, 1994). Secara sederhana, TSP dideskripsikan sebagai permasalahan untuk menetukan sirkuit terpendek yang harus dilalui oleh seorang salesman, yang berangkat dari sebuah kota asal dan menyinggahi setiap kota tepat satu kali kemudian kembali lagi kekota semula. (Admi, Wamiliana dan Yasir; 2008)

Menurut Wardy (2007), *Travelling Salesman Problem* (TSP) merupakan salah satu permasalahan optimasi kombinatorial. Optimasi adalah suatu proses untuk mencapai hasil yang optimal (nilai efektif yang dicapai). Dalam disiplin matematika optimasi merujuk pada studi permasalahan yang mencoba untuk mencari nilai minimal atau maksimal dari suatu fungsi riil. Untuk dapat mencapai nilai optimal baik minimal atau maksimal tesebut, secara sistematis dilakukan pemilihan nilai variabel integer atau riil yang akan memberikan solusi optimal. Nilai optimal adalah nilai yang didapat melalui suatu proses dan dianggap menjadi solusi jawaban yang paling baik dari semua solusi yang ada.

Meskipun telah banyak dilaporkan berbagai metode untuk penyelesaian TSP, namun demikian sampai saat ini diyakini belum ada metode atau algortima yang dapat memberikan solusi eksak untuk TSP yang berukuran besar. Berikut merupakan perkembangan penelitian TSP yang beraitan dengan penelitian kali ini :

Tabel 1. Perkembangan Penelitian TSP

Tahun	Peneliti	Judul
2013	I. Varita, O. Setyawati, dan D. Rahadi	Pencarian Jalur Tercepat Rute Perjalanan Wisata Dengan Algoritma *Tabu Search*
2015	Fatmawati, B. Prihandono, dan E. Noviani	Penyelesaian *Travelling Salesman Problem* Dengan Metode *Tabu Search*
2019	C. Puspitasari, Y. Diah Rosita, dan Y. Prastyaningsih	Optimasi Rute Sales Pengiriman Berdasarkan Jarak dengan Metode Simple Hill Climbing (Studi Kasus CV Maju Jaya)
2020	A. Satriyo	Penerapan Metode *Simple Hill Climbing* Dalam Menentukan Rute Terpendek Pada Pengiriman (Studi Kasus di *Supplier* Hotel)
2021	N. F. Situmorang	Analisis Perbandingan Algoritma *Simple Hill Climbing* dan Algoritma *Tabu Search* Dalam Pemilihan Rute Pendistribusian Uang

Adapun hasil yang dapat peneliti simpulkan berdasarkan penelitian tersebutadalah :

1. Semakin besar jumlah iterasi pada *Tabu Search* akan mendapatkan solusi yang lebih baik.
2. Algoritma *simple hill climbing* bersifat dinamis dimana data yang dicapai berupa masukan, jika penambahan simpul maka rute yang ditempuh akan berbeda dan perhitungan data berbeda akan tetapi tetap memberikan solusi yang optimal.
3. Algoritma *Simple Hill Climbing* membutuhkan waktu lebih cepat dalam proses pengerjaannya dengan jarak pada rute baru yang dihasilkan lebih baik dari solusi awal. Sedangkan Algoritma Tabu Search membutuhkan waktu pengerjaan yang lebih lama, namun Algoritma ini menghasilkan rute baru dengan jarak yang lebih optimal ketimbang Algoritma Simple Hill Climbing.

Menurut Ferdian (2008), *Travelling Salesman Problem* (TSP) adalah mencari sirkuit terpendek pada suatu graf tidak berarah yang berasal dari suatu simpul dengan melewati seluruh simpul dan kembali ke simpul asal. Sirkuit ini disebut juga sirkuit Hamilton. Pembuktian pada graf hamilton terdapat dua buah teorema yaitu Teorema Dirac dan Teorema Ore.

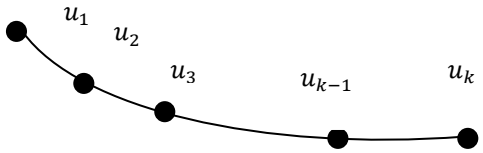
Teorema Dirac :

Syarat cukup (bukan syarat perlu) graf sederhana G dengan orde n ≥3 adalah graf Hamilton bila derajat tiap simpul d(v)≥ $n/2$. (Hasmawati, et al. 2013)

Bukti Teorema Dirac:

Bukti menggunakan induksi matematika atas banyaknya simpul n di G. Jika G mempunyai n = 3 dan d(v) > 3/2 untuk setiap v di G, maka d(v) = 2 yang merupakan graf hamilton. Jadi teorema benar untuk n = 3. Teorema dianggap benar untuk n> 4. Misalkan J adalah jalur terpanjang di G.

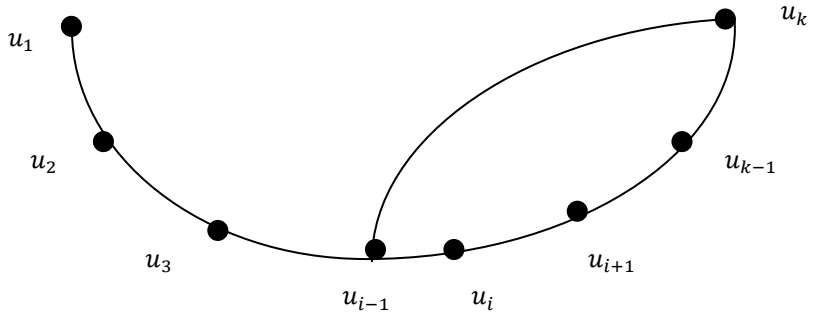
J : $u_1, u_2, \dots, u_k$, seperti pada gambar dibawah ini:



Gambar 7. Pembuktian Teorema Dirac bagian 1

Ini berarti bahwa J memuat simpul paling banyak yang mungkin dimuat oleh J di G. Karena tidak ada lagi jalur di G yang memuat simpul lebih banyak daripada yang dimiliki J, maka setiap simpul yang bertemu dengan u_1 harus berada di J. Selain itu, setiap simpul yang bertemu (atau disebut juga "berdekatan") dengan u_k harus berada di J. Karena paling sedikit berdekatan dengan n/2 simpul yang semuanya berada di J, maka J paling sedikit memuat sebanyak 1 + n/2 simpul. Selanjutnya, haruslah terdapat simpul u_i di J,

dengan $2 < i < k$, sedemikian rupa sehingga u_i berdekatan dengan u_1, dan u_k berdekatan dengan u_{i-1}. Akan tetapi oleh karena paling sedikit ada n/2 simpul-simpul u_i berdekatan dengan simpul u_1 , maka akan ada paling sedikit n/2 simpul-simpul u_{i-1}tidak akan berdekatan dengan u_k. Akibatnya $d(u_k) \leq (n-1) - \frac{n}{2} \leq \frac{n}{2}$, hal ini sesuatu yang tidak mungkin sebab $d(u_k) \geq \frac{n}{2}$. Dengan demikian, terdapat suatu simpul u_i di J sedemikian rupa sehingga $u_1 u_i$ dan $u_{i-1} u_k$ kedua-duanya merupakan rusuk G (periksa Gambar 8). Akibatnya adalah terdapat siklus S: $u_1, u_i, u_{i+1}, \ldots, u_{k-1}, u_{k-2}, \ldots, u_1$ yang memuat semua simpul di J. Jika semua simpul di G telah berada di S, maka S adalah siklus Hamilton dan G adalah graf Hamilton.



Gambar 8. Pembuktian Teorema Dirac bagian 2

Misalkan ada suatu simpul w di G yang tidak berada di S. Karena S memuat paling sedikit 1 + n/2 simpul, sebanyak kurang dari n/2 simpul di G tidak berada di S. Karena $d(w) \geq \frac{n}{2}$, simpul w harus berdekatan dengan suatu simpul u_j di S. Akan tetapi, rusuk uw_j dan S membangun jalur yang banyaknya simpul 1 lebih banyak daripada simpulsimpul yang berada di J, yang tidak mungkin terjadi sebab J memuat simpul-simpul paling banyak. Kontradiksi ini berakibat bahwa S memuat semua simpul di G, sehingga G adalah graf Hamilton. (Terbukti)

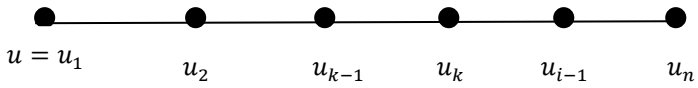
Teorema Ore:

Jika G adalah Graf sederhana dengan n buah simpul (n ≥3) sedemikian sedemikian sehingga d(v)+d(u) $\geq n$ untuk setiap pasang simpul tidak bertetangga u dan v maka G adalah Graf Hamilton. (Hasmawati, et al. 2013)

Bukti Teorema Ore:

Akan dibuktikan G hamilton. Misal G bukan Hamilton yang memenuhi kondisi($d(u) + d(v) \geq n$. Asumsikan bahwa titik di G dengan penambahan himpunan sisi $\{u,v\}$ akan membentuk graf Hamilton, untuk dua titik u,v yang

tidak bertetangga. Karena G bukan Hamilton dan G + { u, v } Hamilton, maka untuk setiap siklus Hamilton di G + { u,v } memuat siklus {u, v}. Sehingga pada G terdapat lintasan Hamilton $u_1, u_2, \dots, u_n$ dengan $u_1 = n$ dan $u_n = v$. .



Sekarang jika setiap $2 < k < n$,$\{u_1, u_k\} \in E(G)$ maka $\{u_{k-1}, u_n\} \in E(G)$. Sebab kalau tidak $u_1, u_k, \dots, u_n, u_{k-1}, u_{k-2}, \dots, u_n$, , adalah siklus Hamilton dari G, sehingga untuk setiap titik yang bertetangga dengan titik v, akan terdapat titik lainnya tidak bertetangga dengan v.

Jadi setiap u_1 bertetangga ke r titik dari $\{u_2, u_3, \dots, u_n\}$ maka paling sedikit r titik, jadi jika $d(v_1) \geq r$. , maka $d(u_n) > (n-1) - d(v_1)$ sehingga $d(v_1) + d(u_n) > n - 1$. Dengan demikian hipotesis $d(v_1) + d(u_n) > n$. (Terbukti)

Dengan terpenuhinya teorema dirac dan ore pada suatu graf, maka dapat disimpulkan graf tersebut memiliki sirkuit hamilton sehingga permasalahan yang akan diangkat nantinya dapat diselesaikan dengan konsep *Travelling Salesman Problem* (TSP).

2.5 Model *Travelling Salesman Problem* (TSP)

Menurut Admi Syarif (2014), Persoalan TSP dapat diformulasikan sebagai masalah integer programming berikut :

Dimisalkan,

$x_{i,j}\begin{cases}=1, & jika\ kota\ dikunjungi\ dari\ kota\ i\ kekota\ j \\ =0, & lainnya\end{cases}$

dan $d_{i,j}$ = jarak dari kota i ke kota j

Bentuk umum TSP adalah sebagai berikut :

$$\textbf{min}\ \ z(x) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{i,j}x_{i,j}\ , i \neq j \qquad = d_{1,2}x_{1,2} + d_{2,3}x_{2,3} + d_{3,4}x_{3,4} + \cdots + d_{n,1}x_{n,1} \ \dots\dots\dots\dots(1)$$

Sehingga:

$$\sum_{j=1;j\neq i}^{n} x_{i,j} = 1, \qquad i = 1,2,\dots,n \qquad x_{1,j} + x_{2,j} + x_{3,j} + x_{4,j} = 1\ \forall\, j = 1,2,\dots,n \ \dots\dots(2)$$

$$\sum_{i=1;i\neq j}^{n} x_{i,j} = 1, \qquad j = 1,2,\dots,n \qquad x_{i,1} + x_{i,2} + x_{i,3} + x_{i,4} = 1\ \forall i = 1,2,\dots,n \ \dots\dots\dots(3)$$

$X_{i,j}$ membentuk sebuah rute.

$X_{i,j} \in \{0,1\}$ seorang sales harus mengunjungi semua kota hanya satu kali.

Keterangan :

n = banyaknya kota dikurangi 1 (karena titik awal dan akhir sama)

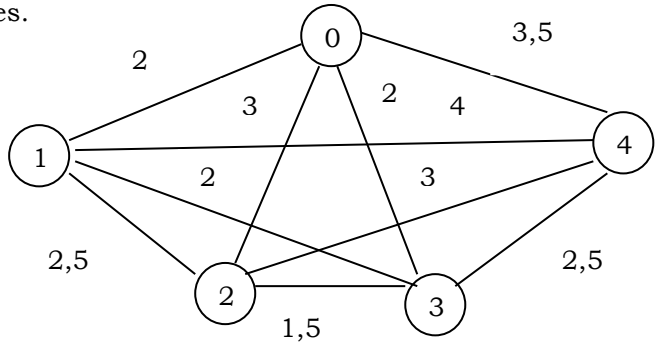
Fungsi tujuan (1) = total jarak atau biaya yang diminimumkan,

Persamaan (2) = bahwa setiap kota (objek) dikunjungi hanya satu kali,

Persamaan (3) = bahwa setiap kota ditinggalkan sekali.

Penjabaran formula dengan contoh kasus:

Berikut merupakan bentuk graf pada suatu rute yang akan dilalui seorang sales dalam mengantarlkan barang ke pelanggannya dengan bobot jarak antar masing-masing kantor/rumah dengan titik 0 sebagai titik awal dan akhir perjalanan sales.



Misalkan untuk salah satu solusi rute yaitu 0-1-2-3-4-0. Total jarak atau biaya yang diminimumkan:

n = 4 (banyaknya kota dikurangi 1)

$d_{i,j}$= bobot jarak dari i ke j

$$\textbf{min}\ \ z(x) = \sum_{i=0}^{4}\sum_{j=0}^{4} d_{i,j}x_{i,j}\ , i \neq j = d_{0,1}x_{0,1} + d_{1,2}x_{1,2} + d_{2,3}x_{2,3} + d_{3,4}x_{3,4} + d_{4,0}x_{4,0}$$

$$= \quad 2x1 + 2{,}5x1 + 1{,}5x + 1 + 2{,}5x1 + 3{,}5x1$$

$$= 12 \text{ Km}$$

Jadi, total dari solusi rute yang dihasilkan adalah 12 Km.

Permasalahan dalam bidang transportasi darat merupakan salah satu penerapan TSP dengan tujuan jarak tempuh yang dilalui dan waktu perjalanan seminimum mungkin. Perjalanan tersebut dapat dimodelkan dalam bentuk graf. Graf adalah himpunan yang terdiri dari simpul dan sisi. Simpul merepresentasikan kota, sisi merepresentasikan jalur yang menghubungkan dua kota, dan bobot merepresentasikan biaya atau jarak tempuh atau waktu perjalanan yang dikeluarkan, yang pada penelitian kali ini peneliti

menggunakan algoritma Tabu Search dan Simple Hill Climbing dalam proses penyelesaian masalah TSP.

2.6 Algoritma *Tabu Search*

Pencarian Tabu pertama kali diusulkan oleh Fred Glover dalam sebuah artikel yang diterbitkan di 1986 [Glover, 1986], Menurut Glover dan Laguna (1997), kata tabu atau "*Taboo*" berasal dari Bahasa Tongan, suatu Bahasa polinesia yang digunakan oleh suku aborigin pulau tonga untuk mengindikasikan suatu hal yang tidak boleh "disentuh" karenakesakralannya. Bahaya yang harus dihindari adalah Metode *Tabu Search* adalah penjadwalannya yang tidak layak, dan terjebak tanpa ada jalan keluar. Menurut Kamus Besar Bahasa Indonesia (KBBI), arti kata tabu adalah hal yang tidak boleh disentuh, diucapkan dan sebagainya karena berkaitan dengan kekuatan suprnatural yang berbahaya (ada risiko kutukan). Arti lainnya dari tabu adalah pantangan.

Menurut Kusuma Dewi dan Purnomo (2005) dalam bukunya yang berjudul "Penyelesaian Masalah Optimasi dengan Teknik Heuristik", Metode *Tabu Search* merupakan metode optimasi yang menggunakan *short-term memory* untuk menjaga agar proses pencarian tidak terjebak pada nilai optimal lokal. Dan menurut Suyanto (2010) Metode *Tabu Search* adalah suatu metode optimasi matematis yang termasuk kedalam kelas *local search*. Metode *Tabu Search* memperbaiki performansi *local search* dengan memanfaatkan penggunaan struktur *memory*. Sebagian solusi yang pernah dibangkitkan ditandai sebagai "*Tabu*" (dalam ejaan lain adalah "*Taboo"* yang berarti suatu yang terlarang), sehingga algoritma *Tabu Search* tidak akan mengunjungi solusi tersebut secara berulang-ulang.

Selain itu menurut Fatmawati, dkk (2015), Metode *Tabu Search* merupakan metode optimasi yang berbasis pada pencarian solusi tetangga (*neighbour solution*) dan memori lokal (*local search*) dengan melakukan move melalui penukaran dua titik. Move yang dimaksud adalah proses pencarian bergerak dari satu solusi kesolusi berikutnya. Konsep dasar dari Metode *Tabu Search* yaitu menuntun setiap tahapannya agar dapat menghasilkan solusi yang paling optimal tanpa terjebak kedalam solusi awal yang ditemukan selama tahapan ini berlangsung.

Algoritma Tabu Search merupakan salah satu aloritma yang berada dalam ruang lingkup metaheuristik. Konsep dasar dari algoritma *Tabu Search* adalah menuntun setiap tahapannya agar dapat menghasilkan fungsi tujuan yang paling optimum tanpa kembali ke dalam solusi awal yang telah ditemukan. Tujuan dari algoritma ini adalah mencegah terjadinya perulangan

dan menemukan solusi yang sama pada suatu iterasi yang akan digunakan lagi pada iterasi selanjutnya (Miswanto, et al. 2018).

Nearest Neighbor (Tetangga Terdekat) dalam metode heuristik digunakan menjadi dasar penentuan rute bagi metode-metode metaheuristik. *Nearest Neighbor* (Tetangga Terdekat) merupakan sebuah metode supervised yang berarti membutuhkan data training untuk mengklasifikasikan objek yang jaraknya paling dekat. Prinsip kerja *Nearest Neighbor* adalah mencari jarak terdekat antara data yang akan di evaluasi dengan n tetangga (neighbor) dalam data pelatihan (Whidhiasih et al., 2013).

Menurut Sulistiono (2015), proses perhitungan Metode *Tabu Search* terdiri dari lima langkah :

1. Mencari solusi dengan algoritma *nearest neighbor* (tetangga terdekat) yang tidak melanggar tabu atau memenuhi kriteria aspirasi sebagai iterasi 0 dan menetapkan solusi dari *nearest neighbor* (tetangga terdekat) sebagai nilai solusi optimum sementara.
2. Mencari solusi alternatif baru dengan menggunakan aturan kombinasi dengan cara menukar 2 titik atau 2 posisi secara berurutan berdasarkan solusi sebelumnya. Memilih solusi terbaik diantara solusi alternatif baru pada tiap iterasi yang akan disimpan sebagai solusi optimum.
3. Memperbarui *Tabu List* dengan memasukkan titik yang telah digunakan pada pertukaran titik dilangkah ketiga.
4. Apabila kriteria pemberhentian dipenuhi maka proses perhitungan *Tabu Search* berhenti dan diperoleh solusi optimum, jika tidak dipenuhi maka proses kembali berulang dimulai pada langkah kedua.

Hal-hal diatas dapat dikatakan sebagai bahan-bahan dasar Metode *Tabu Search* yang nantinya akan digunakan untuk memecahkan masalah rute optimal pada pembahasan selanjutnya.

Adapun keunggulan dari Algoritma Tabu Search sebagai berikut :

1. Menurut Glover (1990), Keunggulan Algoritma *Tabu Search* adalah adanya *tabu list* yang fleksibel sehingga membedakan metode ini dengan metode *Branch and Bound* yang menggunakan struktur memori yang kaku serta Metode *Simulated Anealling* yang tidak menggunakan struktur memori dan tidak menggunakan pembentukan kandidat solusi secara acak. Metode *Tabu Search* menyimpan solusi terbaik serta terus mencari berdasarkan solusi terakhir. Hal ini yang membuat Metode *Tabu Search* menjadi lebih efisien dalam hal usaha dan waktu. Kemampuan metode ini dalam menghasilkan solusi telah dimanfaatkan dalam berbagaimacam permasalahan klasik dan praktis yang salah satunya dalam pemodelan graf.

2. Genreau (1998) mengemukakan bahwa Algortima *Tabu Search* adalah pendekatan yang paling efektif untuk pemecahan masalah penentuan jalur kendaraan. Algoritma *Tabu Search* sangat popular ditahun 90an, dan sampai sekarang tetap menjadi salah satu solusi yang banyak dipakai untuk menyelesaikan permasalahan optimasi.
3. Algoritma *Tabu Search* umumnya tidak menggunakan pembentukan kandidat solusi secara acak sebagaimana *Simulated Anealling* dan *Genetic Algorithm.* Pemilihan kandidat solusi dalam Algoritma *Tabu Search* juga tidak dilakukan secara probabilistik sebagaimana *Ant Colony, System Simulaed Anealling dan Genetic Algorithm.* Karakteristik ini menjadikan solusi yang dihasilkan Algoritma *Tabu Search* akan sama setiap kali dilakukan proses pencarian solusi untuk suatu permasalahan. Karakteristik ini juga menjadi salah satu keunggulan Metode *Tabu Search* dibandingkan dengan metode lainnya . (Alamin, 2009)

2.7 Algoritma *Simple Hill Climbing*

Metode *Simple Hill Climbing* merupakan cabang dari metode *Hill Climbing*. *Hill Climbing* mengembangkan satu node pada suatu waktu yang dimulai dengan node awal(Potdar, G. et al. 2014). *Simple Hill Climbing* adalah salah satu metode dari sekian banyak metode kecerdasan buatan untuk menyelesaikan permasalahan optimasi dengan efisiensi dari segi memori (Sitanggang, D. 2015).

Langkah *Simple Hill Climbing* dimulai dari posisi yang paling kiri setelah menentukan lintasan awal dan melakukan pergantian titik lokasi, lalu membandingkan keadaan *n* titik sekarang dengan satu titik tanpa memperhatikankeadaan titik berikutnya di level yang sama, dan titik awal yang lebih baik akan dipilih menjadi keadaan selanjutnya. Langkah tersebut dilakukan terus menerus hingga tujuan akhir yang paling optimum telah didapatkan (Nana, K, K. et al. 2015).

Berikut ini merupakan langkah-langkah dalam algoritma *Simple Hill Climbing* (Irfan, M. 2017) :

1. Membuat *initial state* dari sembarang kombinasi dari titik-titik/simpul.
2. Evaluasi *initial state*. Jika *initial state* adalah *goal state* maka jadikan *state*ini sebagai solusi. Jika bukan *goal state*, lanjutkan proses dengan *initial state* sebagai *current state*.
3. Ulangi sampai solusi ditemukan atau sampai tidak ada operator baru yang dapat diaplikasikan terhadap *current state*:
   a. Pilih operator yang belum diaplikasikan terhadap *current state* dan aplikasikan operator tersebut sehingga menghasilkan *new state*.
   b. Evaluasi *new state* :

Jika *state* ini merupakan *goal state* maka jadikan *state* ini sebagai solusi.

i. Jika state ini bukan *goal state* tetapi lebih baik dari *current state* maka jadikan *state* ini sebagai *current state* baru.

ii. Jika *state* ini tidak lebih baik dari *current state* maka iterasi dihentikan dan *current state* ditetapkan sebagai solusi optimum.