IMPLEMENTASI ALGORITMA *FLOYD-WARSHALL*

UNTUK MENENTUKAN RUTE TERCEPAT PADA

JALUR DISTRIBUSI JASA PENGIRIMAN BARANG

Skripsi
Disusun untuk melengkapi syarat-syarat
guna memperoleh gelar Sarjana Sains

SRI BASKORO BAGUS PRATIKNO
3125111213

PROGRAM STUDI MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI JAKARTA
2017

LEMBAR PERSETUJUAN HASIL SIDANG SKRIPSI

IMPLEMENTASI ALGORITMA *FLOYD-WARSHALL* UNTUK MENENTUKAN RUTE TERCEPAT PADA JALUR DISTRIBUSI JASA PENGIRIMAN BARANG

Nama : Sri Baskoro Bagus Pratikno

No. Registrasi : 3125111213

	Nama	Tanda Tangan	Tanggal
Penanggung Jawab			
Dekan	: Prof. Dr. Suyono, M.Si.	....	....
	NIP. 19671218 199303 1 005		
Wakil Penanggung Jawab			
Wakil Dekan I	: Dr. Muktiningsih, M.Si.	....	....
	NIP. 19640511 198903 2 001		
Ketua	: Dra. Widyanti Rahayu, M.Si.	....	....
	NIP. 19661103 200112 2 001		
Sekretaris	: Med Irzal, M.Kom.	....	....
	NIP. 19770615 200312 1 001		
Penguji	: Ir. Fariani Hermin, MT.	....	....
	NIP. 19600211 198703 2 001		
Pembimbing I	: Drs. Mulyono, M.Kom.	....	....
	NIP. 19660517 199403 1 003		
Pembimbing II	: Ria Arafiyah, M.Si.	....	....
	NIP. 19751121 200501 2 004		

Dinyatakan lulus ujian skripsi tanggal: 11 Agustus 2017

LEMBAR PENGESAHAN

Dengan ini saya mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta

Nama	:	Sri Baskoro Bagus Pratikno
No. Registrasi	:	3125111213
Jurusan	:	Matematika
Judul	:	Implementasi Algoritma *Floyd-Warshall* Untuk Menentukan Rute Tercepat Pada Jalur Distribusi Jasa Pengiriman Barang

Menyatakan bahwa skripsi ini telah siap diajukan untuk seminar skripsi.

Menyetujui,

Dosen Pembimbing I

Drs. Mulyono, M.Kom.
NIP. 19660517 199403 1 003

Dosen Pembimbing II

Ria Arafiyah M.Si.
NIP. 19751121 200501 2 004

Mengetahui,

Ketua Program Studi Matematika

Dr. Lukita Ambarwati, S.Pd., M.Si.
NIP. 19721026 200112 2 001

ABSTRACT

SRI BASKORO BAGUS PRATIKNO, 3125111213. The implementation of *Floyd-Warshall* Algorithm to Determine The Fastest Route on The Freight Distribution Line. Thesis. Faculty of Mathematics and Natural Science Jakarta State University. 2017.

The problem of freight distribution is one of the important aspects to be considered by every freight forwarder company. This relates to the problem of optimizing freight distribution route from the warehouse to the representative offices and forwarded to the customer. The problem of determining the vehicles route in freight distribution can be modeled as Vehicle Routing Problem (VRP). VRP aims to minimize the total distance or travel time of the vehicle so as to minimize the cost of logistics company with attention to some limitations. One variation of the VRP is the Capacitated Vehicle Routing Problem (CVRP) by adding an identical vehicle capacity constraint, so the length of the vehicle route is limited by the carrying capacity of the used vehicle. This problem can be solved by *Floyd-Warshall* Algorithm. In this thesis will be determined the optimal route on the distribution line of a freight forwarding company using *Floyd-Warshall* Algorithm.

Keywords *: Freight Distribution, Optimal Route, Capacitated Vehicle Routing Problem (CVRP), Algoritma Floyd-Warshall.*

ABSTRAK

SRI BASKORO BAGUS PRATIKNO, 3125111213. Implementasi Algoritma *Floyd-Warshall* Untuk Menentukan Rute Tercepat Pada Jalur Distribusi Jasa Pengiriman Barang. Skripsi. Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta. 2017.

Masalah distribusi barang merupakan salah satu aspek penting yang perlu diperhatikan oleh setiap perusahaan jasa pengiriman barang. Hal ini berkaitan dengan masalah optimasi rute distribusi barang dari gudang menuju ke sejumlah kantor agen perwakilan dan diteruskan menuju pelanggan. Masalah penentuan rute kendaraan dalam mendistribusikan barang dapat dimodelkan sebagai permasalahan Vehicle Routing Problem. VRP bertujuan meminimumkan total jarak ataupun waktu tempuh kendaraan sehingga dapat meminimumkan biaya logistik perusahaan dengan memperhatikan beberapa batasan-batasan. Salah satu variasi VRP adalah Capacitated Vehicle Routing Problem (CVRP) yaitu dengan menambahkan kendala kapasitas kendaraan yang identik, sehingga panjang rute kendaraan dibatasi oleh kapasitas angkut kendaraan yang digunakan. Permasalahan ini dapat diselesaikan oleh Algoritma *Floyd-Warshall.* Pada skripsi ini akan ditentukan rute optimal pada jalur distribusi sebuah perusahaan jasa pengiriman barang menggunakan Algoritma *Floyd-Warshall.*

Kata kunci : Distribusi Barang, Rute Optimal, *Capacitated Vehicle Routing Problem* (CVRP), Algoritma *Floyd-Warshall*.

PERSEMBAHANKU...

"Expect nothing, and you will never be dissapointed"

"Karena sesungguhnya sesudah kesulitan itu ada kemudahan"

"Bertualanglah sejauh mata memandang.
Mengayunlah sejauh lautan terbentang.
Bergurulah sejauh alam terkembang. "

Skripsi ini kupersembahkan untuk Bapak dan Ibu
Seluruh keluarga besar Dharmo Sarjono dan Nadia.
"Terima kasih atas dukungan, do'a, serta kasih sayang kalian".

KATA PENGANTAR

Puji syukur kepada Allah SWT. atas rahmat, hidayah, pengetahuan dan kemampuan-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Implementasi algortima *Floyd-Warshall* untuk menentukan rute tercepat pada jalur distribusi jasa pengiriman barang" yang merupakan salah satu syarat dalam memperoleh gelar Sarjana Jurusan Matematika Universitas Negeri Jakarta.

Skripsi ini berhasil diselesaikan tidak terlepas dari adanya bantuan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis ingin menyampaikan terima kasih terutama kepada:

1. Bapak Drs. Mulyono M.Kom., selaku Pembimbing Akademik merangkap Dosen Pembimbing I dan Ibu Ria Arafiyah M.Si., selaku Dosen Pembimbing II, yang telah meluangkan waktunya dalam memberikan bimbingan, saran, nasehat serta arahan sehingga skripsi ini dapat menjadi lebih baik dan terarah.

2. Ibu Dr. Lukita Ambarwati, S.Pd., M.Si., Selaku Koordinator Program Studi Matematika FMIPA UNJ yang telah banyak membantu penulis.

3. Bapak, Ibu, dan Kakak penulis yang selalu mendukung, memberi motivasi, dan setia membantu penulis dengan penuh cinta dan kasih sayang yang tulus.

4. Nadia Rahmah. Terima kasih banyak karena selalu menjadi sahabat terbaik saat suka maupun duka serta menjadi penyemangat, penasehat, motivator, pembimbing, penjaga, dan pendukung bagi penulis.

5. Mahdhi, Anisa Idam, dan Firdha Defita Sari, selaku teman seperjuangan

penulis dalam mengerjakan skripsi ini. Terimakasih banyak atas bantuannya, masukannya, semangatnya, serta do'anya sehingga penulis dapat menyelesaikan skripsi ini.

6. Adik tingkat di matematika 2012 khususnya Heru Wibowo yang telah memberikan arahan dan motivasi serta membantu penulis.

7. Seluruh Bapak/Ibu dosen atas pengajarannya yang telah diberikan, serta karyawan/karyawati staff FMIPA UNJ yang telah memberikan informasi yang penulis butuhkan dalam menyelesaikan skripsi.

8. Teman-teman rumah yang tidak bisa disebutkan satu per satu. Terima kasih atas segala pengalaman berharga yang telah dilalui, tawa canda kalian merupakan sebuah motivasi tersendiri bagi penulis.

9. Teman-teman dan pihak-pihak yang tidak sempat penulis sebutkan satu per satu. Terima kasih atas segala bantuan dan dukungannya bagi penulis dalam pengerjaan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Masukan dan kritikan akan sangat berarti. Semoga skripsi ini dapat bermanfaat bagi pembaca sekalian.

Jakarta, Agustus 2017

Sri Baskoro Bagus Pratikno

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Kebutuhan pengiriman barang telah menjadi kebutuhan utama setiap individu terutama pengiriman barang yang cepat dan aman. Untuk memenuhi kebutuhan pengiriman barang, saat ini banyak lahir perusahaan jasa pengiriman barang yang terus berkembang dan bersaing untuk merebut pasar. Ada banyak contoh perusahaan di Indonesia yang bergerak pada bidang jasa pengiriman barang atau dokumen, diantaranya PT. Jalur Nugraha Ekakurir (JNE), PT. Pos Indonesia, PT. Global Jet Ekspres (JET), PT. Eka Sari Lorena (ESL) Ekspres, dan PT. Pahala Ekspres, adalah beberapa contoh perusahaan yang bergerak pada bidang jasa pegiriman barang atau dokumen di Indonesia.

Berkembang pesatnya jasa pengiriman barang di Indonesia berbanding lurus dengan banyaknya perusahaan-perusahaan baru yang bergerak di bidang tersebut. Tidak hanya melayani pengiriman barang dalam negeri, perusahaan-perusahaan tersebut juga menyediakan jasa pengiriman barang ke luar negeri. Dalam pengiriman barang kepada *customer*, perusahaan-perusahaan tersebut tentu memiliki beberapa permasalahan yang salah satunya adalah ketepatan waktu tiba barang atau dokumen kepada *customer*. Sebagaimana yang ditulis oleh Donoriyanto dalam "*Analisis Kualitas Pelayanan Jasa Pengiriman Barang dengan metode Servqual dan QFD*" (2012: 10) bahwa ketepatan waktu tiba sebuah barang masih menjadi salah satu faktor ketidakpuasan *customer* pada perusahaan jasa pengiriman barang. Karena persaingan yang semakin ketat, jelas ketepatan waktu tiba barang atau dokumen adalah yang utama. Meminimalkan jarak ataupun waktu tempuh pada jalur pendistribusian barang mungkin menjadi salah satu cara yang cukup efektif untuk mengatasi masalah ketepatan waktu tiba sebuah barang ataupun dokumen. Masalah tersebut dimulai dari menemukan jarak ataupun waktu tempuh yang paling optimal, sehingga mendapat meminimalkan jarak maupun biaya perjalanan yang optimal. Agar perusahaan tersebut tidak mengalami kerugian dalam pendistribusian barang, maka perusahaan tersebut memerlukan metode untuk meminimalkan jarak ataupun waktu tempuh yang akan berkaitan dengan biaya pendistribusian barang. Dengan menggunakan metode dan teori yang berkembang saat ini, berbagai masalah tersebut bisa dicari solusinya. Salah satu teori yang paling terkenal adalah teori graf. Aplikasi dalam teori graf sangat luas karena di dalam teori graf terdapat banyak metode yang bisa digunakan dengan kondisi yang bermacam-macam.

Pencarian lintasan tercepat merupakan suatu masalah graf yang paling banyak dibahas dan dipelajari sejak akhir tahun 1950. Pencarian lintasan tercepat ini telah diterapkan di berbagai bidang untuk mengoptimasi kinerja suatu sistem, baik untuk meminimalkan biaya atau mempercepat jalannya suatu proses. Dalam pencarian rute tercepat, penghitungan dapat dilakukan dengan beberapa macam algoritma. Secara garis besar algoritma penghitungan rute tercepat dibagi menjadi dua kelas berdasarkan metode pemberian labelnya, yaitu algoritma *label setting* dan algoritma *label correcting.* Metode *label setting* menentukan label bobotnya sebagai bobot permanen pada setiap iterasinya, sedangkan metode *label correcting* menentukan label bobotnya sebagai temporal pada setiap iterasi sampai langkah akhir. Ketika semua node telah melewati proses pemeriksaan, maka labelnya akan ditentukan sebagai permanen. Algoritma *Dijkstra* adalah salah satu algoritma penghitungan rute tercepat kelas *label setting*, sedangkan pada kelas *label correcting* terdapat algoritma *Floyd* dan algoritma *Two-Queues*. Algoritma yang digunakan untuk pencarian jarak tercepat adalah algoritma *Floyd-Warshall*.

Ada beberapa model permasalahan pencarian rute tercepat yang bisa diselesaikan oleh algoritma *Floyd-Warshall*, salah satunya adalah *Vehicle Routing Problem* (VRP). *Vehicle Routing Problem* merupakan masalah penentuan rute tercepat kendaraan dalam pendistribusian barang dari satu depot atau lebih ke sejumlah konsumen di lokasi yang berbeda dengan permintaan yang telah diketahui dan memenuhi sejumlah kendala. Tujuan dari *Vehicle Routing Problem* adalah mengantarkan barang ke konsumen dengan rute yang tercepat dan meminimalisasi jumlah kendaraan yang digunakan untuk keluar-masuk depot. Beberapa jenis permasalahan VRP berdasarkan kendalanya salah satunya adalah *Capacitated Vehicle Routing Problem* (CVRP) dengan kendala kapasitas setiap kendaraan terbatas. Contoh kasus dalam kehidupan sehari-hari yang dapat dimodelkan dalam permasalahan CVRP antara lain distribusi air minum, distribusi surat kabar, pelayanan jasa kurir, jasa ojek, penentuan rute bus sekolah, dan lain sebagainya, Sebagaimana yang ditulis oleh Iskandar dalam "*Model Optimasi Vehicle Routing Problem dan Implementasinya*" (2010: 38) bahwa Masalah distribusi barang dapat diformulasikan dalam model *Capacitated Vehicle Routing Problem*. Dapat disimpulkan bahwa CVRP dianggap cukup optimal dalam permasalahan distribusi barang dibanding model lain seperti *Chinnese Postman Problem*, karena mempertimbangkan kapasitas angkut kendaraan dalam permasalahannya. Dengan Menggunakan algoritma *Floyd-Warshall*, algoritma tersebut akan menghitung setiap solusi sampai didapat sebuah solusi yaitu rute yang optimal.

Pada skripsi ini dilakukan studi kasus pencarian rute tercepat untuk jalur distribusi jasa pengiriman barang. Ide dari penelitian ini berawal dari masalah ketepatan waktu tiba sebuah barang untuk sampai di tangan *customer*. Faktor kecepatan distribusi para kurir dari gudang menuju kantor agen suatu perusahaan jasa pengiriman barang adalah hal terpenting untuk mencegah menumpuknya barang yang masuk. Pencarian rute tercepat ini memfokuskan untuk mencari lokasi per agen berdasarkan lokasinya yang disimbolkan dengan sebuah *verteks* menggunakan implementasi algortima *Floyd-Warshall* untuk menentukan rute tercepat pada jalur distribusi jasa pengiriman barang dengan metode *Capacitated Vehicle Routing Problem* (CVRP)

1.2 Perumusan Masalah

Permasalahan yang akan diajukan dalam skripsi ini adalah::

1. Apakah masalah penentuan rute tercepat pada model Capacitated *Vehicle Routing Problem* (CVRP) dapat diselesaikan dengan Algoritma *Floyd-Warshall*?

2. Apakah rute yang dihasilkan oleh algoritma *Floyd-Warshall* lebih cepat dibandingkan dengan rute sebelumnya?

1.3 Pembatasan Masalah

Ruang lingkup yang akan dibahas disini adalah:

1. Pada penelitian ini dilakukan studi kasus pencarian rute tercepat untuk jalur pendistribusian barang dari gudang menuju kantor kantor perwakilan agen JET.

2. Mencari penyelesaian masalah lintasan tercepat dengan meminimumkan waktu tempuh.

3. Tidak mempertimbangkan kondisi kemacetan dan *traffic light* yang ada pada kondisi jalan semestinya.

1.4 Tujuan Penulisan

Tujuan dari penelitian ini yaitu:

1. Merepresentasikan Algoritma *Floyd-Warshall* ke dalam permasalahan pencarian rute tercepat.
2. Menyelesaikan permasalahan Capacitated *Vehicle Routing Problem* (CVRP) dengan Algoritma *Floyd-Warshall* pada penentuan rute tercepat jalur distribusi jasa pengiriman barang.

1.5 Manfaat Penulisan

Adapun manfaat dari penelitian ini, yaitu:

1. Bagi Penulis:

   Membantu penulis untuk mengetahui bagaimana penentuan rute tercepat pada jalur distribusi sebuah jasa pengiriman barang menggunakan algoritma *Floyd-Warshall*

2. Bagi Perusahaan:

   Menjadi referensi yang berkaitan dengan teori graf dalam menyelesaikan masalah rute tercepat dalam mendistribusikan barang menuju kantor perwakilan agen.

3. Bagi Mahasiswa:

   Memberikan informasi tentang algoritma *Floyd-Warshall* sekaligus menjadi bahan acuan dalam pengaplikasian teori graf selanjutnya.

BAB II

LANDASAN TEORI

2.1 Teori Graf

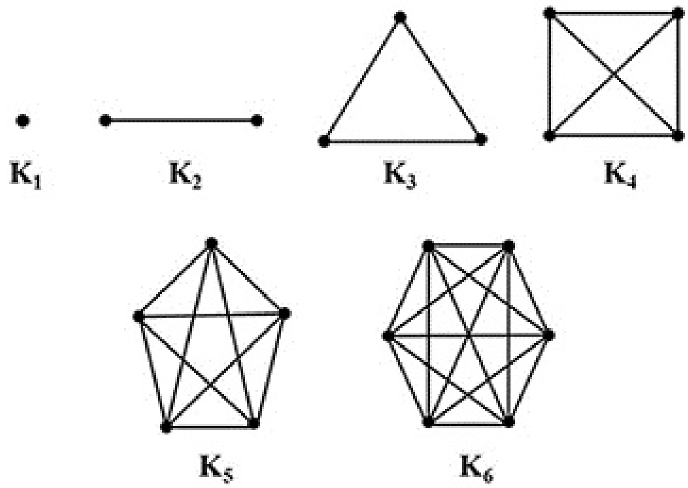
Teori Graf adalah cabang dari matematika diskrit yang mempelajari sifat-sifat graf. Secara informal, suatu graf adalah himpunan benda-benda yang disebut simpul (*vertex* atau *node*) yang terhubung oleh sisi (*edge*). Biasanya graf digambarkan sebagai kumpulan titik-titik aray simpul yang dihubungkan oleh garis-garis atau sisi. Suatu sisi dapat menghubungkan suatu simpul dengan simpul yang sama. Sisi yang demikian dinamakan gelang (*loop*). Teori Graf banyak digunakan sebagai alat bantu untuk menggambarkan atau menyatakan suatu persoalan agar lebih mudah dimengerti dan diselesaikan. Suatu Graf $G(V, E)$ didefinisikan sebagai pasangan himpunan (V, E) dimana V adalah himpunan berhingga dan tidak kosong dari simpul simpul (*vertex*) dan E adalah himpunan berhingga dari busur-busur (*edges*). Biasanya graf ditulis sebagai G tanpa himpunan V dan E nya dengan tujuan untuk mempersingkat.

2.1.1 Jenis-jenis graf

Teori Graf dapat dibagi dalam beberapa jenis. Hal ini tergantung pada bagaimana kekhasan dari graf tersebut. Berikut beberapa jenis graf yang cukup banyak ditemui dalam penerapan teori graf.

- **Graf Lengkap**

Misalkan $G = (V, E)$ adalah sebuah graf sederhana. Jika untuk setiap pasangan *verteks* v_i dan v_j di G terdapat sebuah *edge* yang menghubungkannya, maka G disebut graf lengkap. Sebuah graf lengkap sering juga disebut sebagai graf universal. Graf lengkap dengan n *verteks* dinotasikan dengan K_n.

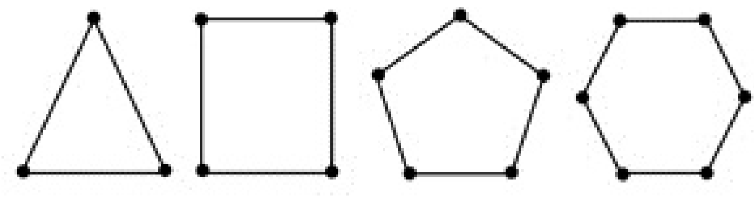


Gambar 2.1: Contoh Graf Lengkap

Banyaknya *edge* dalam graf lengkap adalah $(n(n-1))/2$

- **Graf Lingkaran**

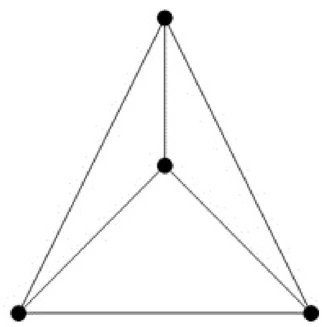
  Graf lingkaran adalah graf sederhana yang setiap *verteks*nya berderajat dua. Graf lingkaran dengan n *verteks* dilambangkan dengan C_n.



Gambar 2.2: Contoh Graf Lingkaran

- **Graf Teratur**

  Sebuah graf disebut graf teratur jika semua *verteks*nya berderajat sama. Apabila derajat setiap *verteks* adalah r , maka graf tersebut disebut sebagai graf teratur derajat r.



Gambar 2.3: Contoh Graf Teratur

- **Graf Bipartite**

  Sebuah graf G disebut graf bipartit jika $V(G)$ (himpunan *verteks* graf G) dapat dipartisi menjadi dua himpunan bagian X dan Y, sedemikian sehingga setiap *edge* pada G menghubungkan sebuah *verteks* di X ke sebuah *verteks* di Y. Apabila G sederhana dan bipartit dengan partisi (X, Y) sedemikian sehingga setiap *verteks* di X bertetangga dengan setiap *verteks* di Y , maka G disebut graf bipartite lengkap, dinotasikan dengan $K_(m, n)$ dengan m dan n adalah banyaknya *verteks* di kedua partisi tersebut.

- **Graf Bagian (*subgraph*)**

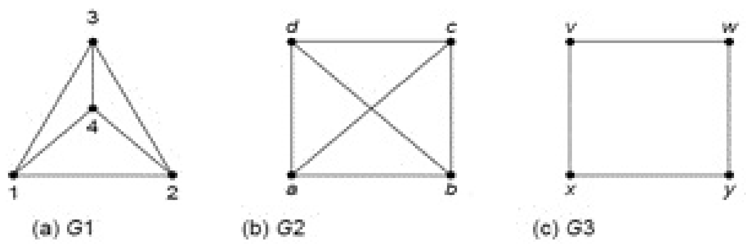
  Sebuah graf K disebut graf bagian (subgraph) dari graf G , dinotasikan $K \subseteq G$, jika $V(K) \subseteq V(G)$ dan $E(K) \subseteq E(G)$. Graf bagian dapat diperoleh melalui suatu operasi penghapusan *verteks* atau penghapusan *edge* pada sebuah graf.

- **Graf Komplemen**

  Komplemen dari sebuah graf G , dinotasikan G', adalah sebuah graf dengan himpunan *verteks* yang sama seperti dalam G dan dengan sifat bahwa dua *verteks* di G bertetangga jika dan hanya jika dua *verteks* yang sama dalam G' tidak bertetangga.

- **Graf Isomorfik**

  Sebuah graf G disebut isomorfik dengan graf H jika terdapat pemetaan satu-satu ϕ (yang disebut isomorfisme dari $V(G)$ ke $V(H)$ sedemikian sehingga ϕ mempertahankan ketetanggaan. Jadi, $(u, v) \in E(G)$ jika dan hanya jika $\phi(u), \phi(v)) \in E(H)$. Jika G isomorfik dengan H , kita tulis $G \cong H$.



Gambar 2.4: Graf Isomorfik. $G1$ Isomorfik dengan $G2$ tetapi tidak dengan $G3$

- **Graf Terhubung**

  Setiap graf G terdiri atas beberapa graf bagian. Komponen graf adalah jumlah maksimum graf bagian dalam sebuah graf G. Sebuah graf disebut terhubung (*connected*) jika graf tersebut hanya terdiri atas satu bagian (satu komponen).

2.1.2 Keterhubungan Suatu Graf (*Connectivity Of Graph*)

Pada kehidupan sehari-hari, seseorang dapat berjalan (berpindah) dari suatu tempat ke tempat yang lain. Dalam hal ini ada tiga kemungkinan yang bisa terjadi, yaitu:

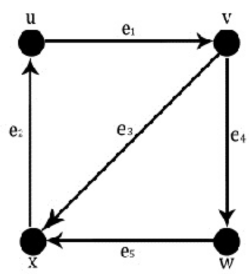
- Orang tersebut berjalan secara bebas, artinya dia dapat melewati sebuah jalan atau mengunjungi sebuah tempat tertentu lebih dari satu kali.
- Orang tersebut melewati suatu jalan tertentu hanya satu kali saja.
- Orang tersebut melewati tempat tertentu hanya satu kali saja.

Pada bahasa teori graf, ketiga kemungkinan perjalanan tersebut, membawa pada tiga konsep yang saling terkait, yaitu jalan (*walk*), jejak (*trail*), dan lintasan (*path*).

- ***Walk***

  Walk W pada suatu graf G adalah barisan berhingga, $W = v_i e_j v_{i+1} e_{j+1} ... e_k v_m$ atau $W = v_i - v_{i+1} - ... - v_m$ yang dimulai dari suatu *verteks* dan berakhir pada suatu *verteks* juga, sehingga setiap *edge* didalam barisan harus incident dengan *verteks* sebelum dan sesudahnya.

  Walk berarah pada suatu digraf D adalah walk yang sesuai arah *edge*nya atau tidak berlawanan arah.

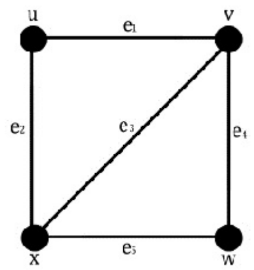


Gambar 2.5: Contoh *walk* berarah

Ilustrasi *walk* berarah pada suatu digraf D bisa dilihat pada gambar 2.5. $W = ue_1ve_4w$ adalah *walk* berarah.

- ***Trail***

  Jejak (*trail*) pada suatu graf adalah *walk* dengan semua *edge* dalam barisannya tidak berulang.



Gambar 2.6: Lintasan

Ilustrasi *trail* yang terlihat pada gambar 2.6 adalah ilustrasi dengan $T = ue_2xe_5we_4v$

Trail berarah pada suatu digraf adalah *walk* berarah dengan semua sisi dalam barisannya tidak terulang. Ilustrasi *trail* berarah bisa dilihat pada gambar 2.5, $T = ue_1ve_4we_5x$ adalah *trail* berarah.

- ***Path***

  Lintasan (*path*) pada suatu graf adalah walk yang setiap *verteks* pada barisannya, hanya muncul satu kali. Ilustrasi *path* bisa dilihat pada gambar 2.6, $P = ue_1ve_4we_5x$ adalah *path* pada graf G.

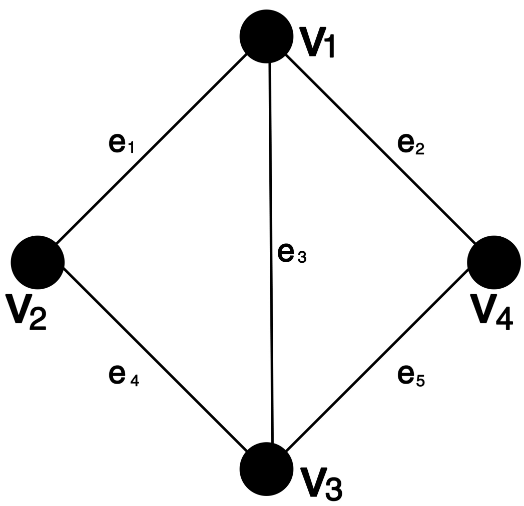
  Path berarah pada suatu digraf adalah *walk* berarah dengan semua *verteks* dalam barisannya tidak berulang. Ilustrasi *path* berarah bisa dilihat pada gambar 2.5, $P = ue_1ve_4we_5x$ adalah *path* berarah.

2.2 Terminologi Dasar Graf

Dalam pembahasan mengenai graf biasanya sering menggunakan terminologi (istilah) yang berkaitan dengan graf. Berikut ini terminologi (istilah) yang berkaitan dengan graf yang akan digunakan dalam skripsi ini, yang dirujukkan dari (Munir, 2005: 364-376).

2.2.1 Bertetangga (*Adjacent*)

Dua buah *verteks* pada graf tak berarah G dikatakan bertetangga bila keduanya terhubung langsung dengan sebuah *edge*. Dengan kata lain, v_i bertetangga dengan v_j jika (v_i, v_j) adalah sebuah *edge* pada graf G.



Gambar 2.7: Contoh Graf G_1

Pada gambar 2.7, verteks v_1 betetangga dengan verteks v_2, v_3 dan v_4. Verteks v_2 bertetangga dengan v_1 dan v_4, tetapi tidak bertetangga dengan v_3

2.2.2 Bersisian (*Incidendt*)

Untuk sembarang *edge* $e = (v_j, v_k)$, *edge* e dikatakan bersisian dengan verteks v_j dan verteks v_k.

Pada gambar 2.7, *edge* e_1 bersisian dengan verteks v_1 dan verteks v_2 *edge* e_5 bersisian dengan verteks v_3 dan verteks v_4, tetapi tidak bersisian dengan v_2.

2.2.3 Derajat(*Degree*)

Derajat suatu *verteks* pada graf tak berarah adalah jumlah *edge* yang bersisian dengan *verteks* tersebut. Pada graf berarah, derajat *verteks* v dinyatakan dengan $d_{in}(v)$ dan $d_{out}(v)$, yang dalam hal ini:

$d_{in}(v)$ = derajat masuk atau jumlah *verteks* yang masuk ke *verteks* v

$d_{out}(v)$ = derajat keluar atau jumlah *verteks* yang keluar ke *verteks* v

Dan

$$d(v) = d_{in}(v) + d_{out}(v)$$

Dalam hal ini $d(v)$ menyatakan derajat *verteks*.

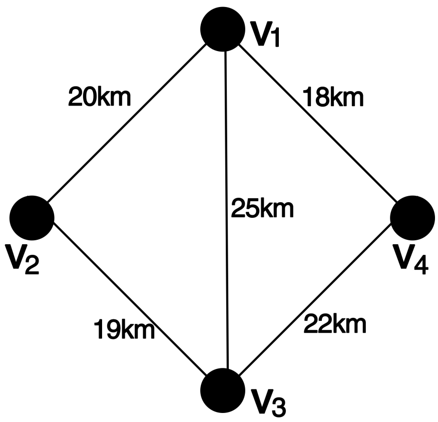
Teorema 2.2.1. Jumlah derajat simpul dalam suatu graf G adalah dua kali banyaknya rusuk atau

$$\sum_{n=1}^{p} d(v_1) = 2q$$

Bukti: Misalkan graf G terdiri satu rusuk, berarti G memiliki dua simpul yang masing-masing berderajat satu, sehingga jumlah derajat simpul dalam G adalah dua. Karena setiap rusuk menghubungkan dua simpul, maka banyaknya rusuk akan menambah jumlah derajat simpul dalam G adalah dua. Ini berarti jumlah derajat simpul dalam G adalah dua kali jumlah rusuk.

2.2.4 Graf Berbobot (*Weighted Graph*)

Graf berbobot adalah graf yang setiap sisinya diberikan sebuah harga (bobot). Bobot pada setiap sisi dapat menyatakan jarak antara dua buah kota, biaya perjalanan, waktu tempuh, ongkos produksi, dan sebagainya.



Gambar 2.8: Contoh Graf Berbobot

Dalam skripsi ini, bobot pada pada setiap graf menyatakan jarak antara dua buah tempat dalam kilometer (km).

2.3 Representasi Graf

Pada penjelasan sebelumnya, graf ditampilkan dengan cara menggambarkannya. Namun apabila graf hendak diproses dengan program komputer, maka graf harus direpresentasikan di dalam memori. Ada beberapa metode yang dapat digunakan dalam merepresentasikan graf, berikut ini adalah metode yang dapat dgunakan dalam merepresentasikan graf:

1. Matriks Ketetanggaan (*Adjacency Matrix*)

   Misalkan $G = (V, E)$ merupakan suatu graf dengan n *verteks*, $n > 1$. Maka, matriks ketetanggaan A dari G adalah matriks nxn dimana $A = [a_{ij}]$, untuk hal ini berlaku $A = [a_{ij}]$ menjadi 1 bila *verteks* i dan j bertetangga dan $A = [a_{ij}]$ menjadi 0 bila verteks i dan j tidak bertetangga.

Jumlah elemen matriks bertetanggaan untuk graf dengan n *verteks* adalah n^2. Jika tiap elemen membutuhkan ruang memori sebesar p, maka ruang memori yang diperlukan seluruhnya adalah pn^2. Pada graf berbobot, a_{ij} menyatakan bobot tiap sisi yang menghubungkan *verteks* i dengan *verteks* j. Bila tidak ada sisi dari *verteks* i ke *verteks* j atau dari *verteks* j ke *verteks* i, maka, a_{ij} diberi nilai tak berhingga (∞).

Bentuk matriks ketetanggaan dari graf pada gambar 2.7 adalah

$$\begin{array}{cc} & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{bmatrix} \infty & 1 & 1 & 1 \\ 1 & \infty & 1 & \infty \\ 1 & \infty & \infty & 1 \\ 1 & \infty & 1 & \infty \end{bmatrix} \end{array}$$

2. Matriks Insiden (*Incidency Matrix*)

   Matriks insiden menyatakan kebersisian *verteks* dengan *edge*. Misalkan $G = (V, E)$ adalah graf dengan n *verteks* dan m *edge*, maka matriks kebersisian A dari G adalah matriks berukuran mxn dimana $A = [a_{ij}]$, $[a_{ij}]$ menjadi 1 bila *verteks* i dan *edge* j bersisian dan $[a_{ij}]$ menjadi 0 bila *verteks* i dan *edge* j tidak bersisian.

Berikut adalah matriks insiden untuk graf pada gambar 2.7.

$$\begin{array}{c c} & \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Pada matriks diatas, sebuah kolom e_7 dapat diwakilkan sebagai *loop*. Pada sebuah graf tanpa *loop*, masing-masing kolom mempunyai dua entri 1, dan jumlah dari sebuah baris menyatakan derajat dari *verteks* yang didefinisikan dengan baris tersebut.

2.4 Lintasan dan Sirkuit Hamilton

Lintasan Hamilton ialah lintasan yang melalui tiap *verteks* di dalam graf tepat satu kali. Bila lintasan itu kembali ke *verteks* asal membentuk lintasan tertutup (sirkuit), maka lintasan tertutup itu dinamakan sirkuit Hamilton. Dengan kata lain, sirkuit Hamilton adalah sirkuit yang melalui tiap *verteks* di dalam graf tepat satu kali, kecuali *verteks* asal (sekaligus *verteks* akhir) yang dilalui dua kali. Graf yang memiliki sirkuit Hamilton dinamakan graf Hamilton, sedangkan graf yang hanya memiliki lintasan Hamilton disebut graf semi-Hamilton.

Teorema 2.4.1. Misalkan G merupakan graf sederhana dengan orde $n \geq 3$. Jika

$$d(u) + d(v) \geq n$$

untuk setiap pasangan titik tak bertetangga u, v pada G, maka G adalah graf

Hamilton.

Bukti: Asumsikan bahwa teorema tersebut tidak benar. Andaikan G bukan graf Hamilton, dan untuk setiap 2 titik tak bertetangga u dan v pada G , maka $G + uv$ adalah graf Hamilton. Diketahui bahwa terdapat graf non Hamilton maksimal G berorde $n \geq 3$. Karena G bukan graf Hamilton, maka bukan G merupakan graf lengkap.

Misalkan u dan v merupakan 2 titik tak bertetangga pada G . Maka $G + uv$ adalah graf Hamilton, dan dengan kata lain, setiap siklus Hamilton pada $G + uv$ memuat sisi uv. Maka terdapat sebuah lintasan $u - v$ yaitu $u = u_1, u_2, ..., u_n = v$ pada G yang memuat semua titik pada G.

Jika $u_1u_i \in E(G)$, $2 \leq i \leq n$, maka $u_{i-1}u_n \notin E(G)$. Sebaliknya, $u_1, u_n, u_{i+1}, ..., u_n, u_{i-1}, u_{i-2}, ..., u_1$ merupakan siklus Hamilton pada G. Karena untuk setiap titik pada $u_2, u_3, ..., u_n$ yang bertetangga dengan u_1 terdapat sebuah titik pada $u_1, u_2, ..., u_{n-1}$ yang tidak bertetangga dengan u_n, maka $d(u_n) \leq (n - 1) - d(u_1)$, sehingga

$$d(u) + d(v) \leq n - 1$$

Hal ini kontradiksi, maka G merupakan graf Hamilton

Teorema 2.4.2. Jika G merupakan graf sederhana dengan n titik dimana $n \geq 3$ dan $d(v) \geq \frac{n}{2}$ untuk setiap titik v pada G, maka G merupakan graf Hamilton

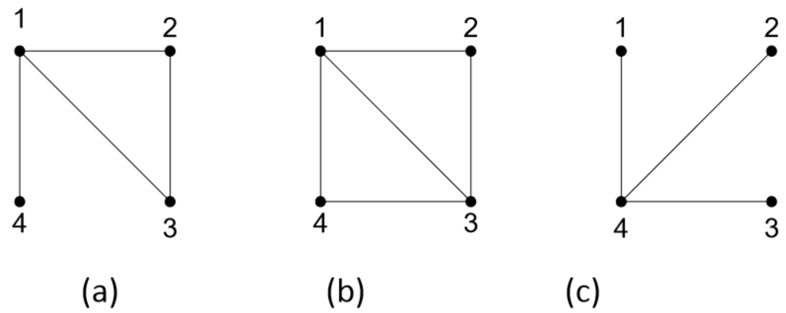
Bukti: Ambil sebarang titik p dan q di G. Karena $\forall v \in G$, $d(v) \geq \frac{n}{2}$, maka

$$d(p) + d(q) \geq \frac{n}{2} + \frac{n}{2} = \frac{2n}{2} = n$$

Menurut Teorema 2.4.1, maka G adalah graf Hamilton.

Teorema 2.4.3. Setiap graf lengkap adalah graf Hamilton.

Bukti: Misalkan setiap verteks diberi penomoran, $v_1, v_2, v_3, ..., v_n$. Untuk setiap graf lengkap pasti selalu memiliki *edge* yang akan menghubungkan setiap *verteks* dalam graf tersebut. Maka kita bisa memulai menyelesaikan graf hamilton dimulai dengan v_1 lalu ke v_2 lalu ke v_3 sampat v_n. Maka ini didapatkan sebuah graf hamilton.



Gambar 2.9: Contoh Graf Hamilton

Keterangan:

a Graf yang memiliki lintasan Hamilton (misal: 3, 2, 1, 4)

b Graf yang memiliki sirkuit Hamilton (1, 2, 3, 4, 1)

c Graf yang tidak memiliki lintasan atau sirkuit Hamilton

2.5 *Vehicle Routing Problem*

2.5.1 Pengertian

Vehicle Routing Problem (VRP) merupakan permasalahan dalam sistem distribusi yang bertujuan untuk membuat suatu rute yang optimal, dengan sekelompok kendaraan yang sudah diketahui kapasitasnya, agar dapat memenuhi permintaan konsumen dengan lokasi dan jumlah permintaan yang telah diketahui. Untuk mencapai tujuan tersebut perlu diperhatikan beberapa batasan yang harus dipenuhi, yaitu setiap kendaraan yang akan mendistribusikan barang ke konsumen harus memulai rute perjalanan dari titik awal (gudang/depot), setiap *verteks* lain hanya boleh dilayani satu kali oleh satu kendaraan, setiap *verteks* mempunyai *demand* (permintaan) yang harus dipenuhi, dan diasumsikan permintaan tersebut sudah diketahui sebelumnya. Setiap kendaraan memiliki batasan kapasitas tertentu, artinya setiap kendaraan akan melayani pelanggan sesuai kapasitasnya. Untuk lebih jelasnya, Berikut adalah kendala atau batasan yang harus di penuhi dalam *Vehicle Routing Problem* (VRP):

a Rute kendaraan dimulai dari depot dan berakhir di depot.

b Masing-masing konsumen harus dikunjungi sekali dengan satu kendaraan.

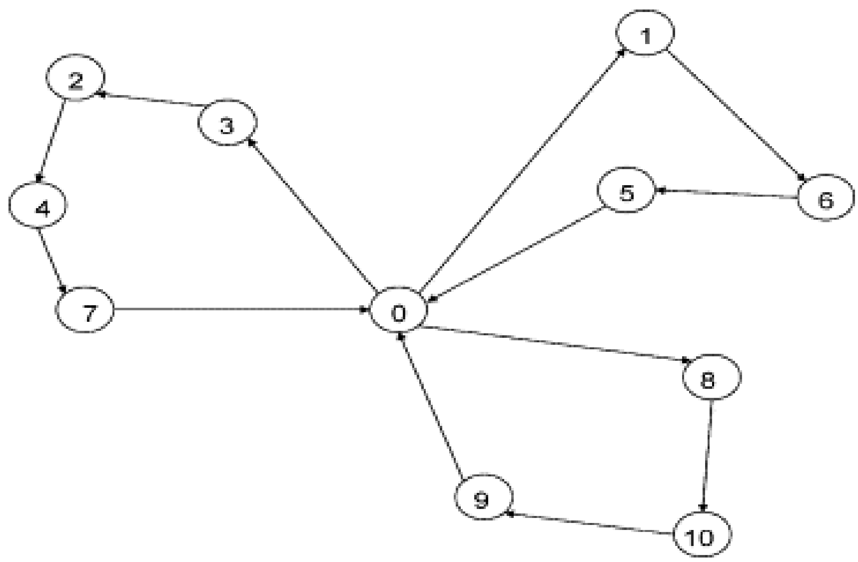
c Kendaraan yang digunakan adalah homogen dengan kapasitas tertentu, sehingga permintaan konsumen pada setiap rute yang dilalui tidak boleh melebihi kapasitas kendaraan.

d Jika kapasitas kendaraan sudah mencapai batas, maka konsumen berikutnya akan dilayani oleh shift berikutnya.

Gambar 2.10 menunjukan solusi dari sebuah permasalahan VRP dalam bentuk graf. Pada gambar tersebut, *verteks* 0 *verteks* awal (gudang/depot), dan *verteks* 1-10 melambangkan konsumen.

Terdapat empat tujuan utama *Vehicle Routing Problem*, yaitu:

a Meminimalkan waktu dan biaya yang dikeluarkan.



Gambar 2.10: Contoh Solusi Dari *Vehicle Routing Problem*

b Meminimalkan jumlah kendaraan.

c Menyeimbangkan rute, untuk waktu perjalanan dan muatan kendaraan.

d Meminimalkan penalti akibat *service* yang lambat yang berakibat pada kepuasan konsumen.

2.5.2 *Capacitated Vehicle Routing Problem*

Masalah yang berkaitan dengan pencarian rute tercepat untuk kendaraan dari satu depot atau gudang dengan kapasitas tertentu untuk melayani sejumlah konsumen sesuai dengan permintaannya masing-masing disebut *Vehicle Routing Problem* (VRP). Dalam masalah VRP ini, setiap *verteks* diasosiasikan sebagai lokasi konsumen dan tiap kendaraan yang digunakan untuk mengunjungi sejumlah konsumen memiliki kapasitas tertentu. Total jumlah permintaan pelanggan dalam suatu rute tidak melebihi kapasitas kendaraan yang ditugasi melayani rute tersebut dan setiap pelanggan dikunjungi hanya satu kali oleh satu kendaraan. Pada masalah VRP juga terdapat suatu depot atau gudang di mana tiap kendaraan harus berangkat dan kembali ke depot atau gudang tersebut. Permasalahan VRP bertujuan meminimalkan total jarak tempuh kendaraan atau total biaya dari setiap rute perjalanan, selain itu bisa juga bertujuan meminimalkan banyaknya kendaraan yang digunakan.

Capacitated Vehicle Routing Problem (CVRP) merupakan salah satu variasi dari masalah VRP dengan penambahan kendala kapasitas kendaraan yang identik. Setiap kendaraan yang melayani konsumen disyaratkan memiliki batasan kapasitas sehingga banyaknya konsumen yang dilayani oleh setiap kendaraan dalam satu rute bergantung pada kapasitas kendaraan. Permasalahan CVRP bertujuan meminimumkan total waktu tempuh rute perjalanan kendaraan dalam mendistribusikan barang dari tempat produksi yang dinamakan dengan depot atau gudang ke sejumlah konsumen.

Kara *et al.* (2004) mendefinisikan *Capacitated Vehicle Routing Problem* (CVRP) sebagai suatu graf berarah $G = (V, A)$ dengan $V = \{v_0, ..., v_n\}$ adalah himpunan *verteks* v_0 menyatakan depot atau gudang sebagai tempat kendaraan memulai dan mengakhiri perjalanan dan $\{v_1, ..., v_n\}$ menyatakan konsumen atau *verteks* yang harus dikunjungi tepat satu kali. Sedangkan $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ adalah himpunan sisi yang menghubungkan antar simpul (*edge*). Setiap *edge* memiliki permintaan (*demand*) sebesar q_i dengan q_i adalah *integer* positif. Himpunan $K = \{k_1, k_2, ..., K_n\}$ merupakan kumpulan kendaraan yang homogen dengan kapasitas yang identik yaitu Q, sehingga panjang setiap rute dibatasi oleh kapasitas kendaraan. Setiap sisi (v_i, v_j) memiliki waktu tempuh Wt_{ij} yaitu jarak dari simpul i ke simpul j dengan $Wt_{ij} = Wt_{ji}$ dan $Wt_{ii} = 0$.

Permasalahan dari CVRP adalah menentukan himpunan dari K rute kendaraan yang memenuhi kondisi berikut:

a Setiap rute berawal dan berakhir di depot atau gudang, b Setiap konsumen harus dilayani tepat satu kali oleh satu kendaraan,

c Total permintaan konsumen dari setip rute tidak melebihi kapasitas kendaraan, dan

d Total waktu tempuh dari semua rute diminimumkan.

Permasalahan tersebut kemudian diformulasikan ke dalam model matematika dengan tujuan meminimumkan total jarak tempuh perjalanan kendaraan. Variable x_{ijk} adalah variabel keputusan yang bernilai 1 jika (v_i, v_j) merupakasan solusi dari masalah CVRP dan bernilai 0 jika bukan solusi, dan variabel u_{ik} merupakan *integer* yang dihubungkan dengan setiap konsumen v_i. Variabel keputusan x_{ijk} dan x_{jik} hanya akan terdefinisi jika $q_i + q_j < Q$. Adapun formulasinya adalah sebagai berikut:

$$\min z = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(Wt_{ij} \sum_{k=1}^{m} x_{ijk} \right)$$

dengan variabel keputusan sebagai berikut:

a Variabel X_{ijk}, $\forall i, j \in N$, $\forall k \in K$, $i \neq j$

Variabel X_{ijk} merepresentasikan ada atau tidaknya perjalanan dari konsumen ke-i ke konsumen ke-j oleh konsumen ke-k

$$X_{ijk} = \begin{cases} 1, & \text{jika ada perjalanan dari konsumen } i \text{ ke konsumen } j \\ & \text{oleh kendaraan } k. \\ 0, & \text{jika tidak ada perjalanan dari konsumen } i \text{ ke konsumen } j \\ & \text{oleh kendaraan } k. \end{cases}$$

b Variabel T_{ik}, T_{0k}, dan s_{ik}, $\forall i \in N, \forall k \in K$

Variabel T_{ik} menyatakan waktu dimulainya pelayanan pada konsumen ke-i oleh kendaraan ke-k, T_{0k} menyatakan waktu saat kendaraan ke-k meninggalkan depot dan kembali ke depot, sedangkan s_{ik} menyatakan lamanya pelayanan konsumen ke-i.

c Variabel Y_{ik} dan q_j, $\forall i, j \in N, \forall k \in K$

Variabel $Y_i k$ menyatakan kapasitas total dalam kendaraan ke-k setelah melayani konsumen ke-i, sedangkan q_j menyatakan banyaknya permintaan konsumen ke-j

Adapun kendala dari permasalahan *Capacitated Vehicle Routing Problem* adalah sebagai berikut:

a Setiap konsumen hanya dikunjungi tepat satu kali oleh kendaraan yang sama.

$\sum_{k=1}^{m} \sum_{j=1}^{n} \sum_{i=1}^{n} X_{ijk} = 1, \forall i, j \in N, \forall k \in K$

b Total jumlah permintaan konsumen dalam satu rute tidak melebihi kapasitas kendaraan yang melayani rute tersebut. Jika ada lintasan dari i ke j, maka:

$Y_{ik} + q_j = Y_{jk}$, $\forall i, k \in N$, $\forall k \in K$

$Y_{jk} \leq Q$, $\forall j \in N$, $\forall k \in K$

c Jika ada perjalanan dari konsumen ke-i ke konsumen ke-j, maka waktu tempuh memulai pelayanan di konsumen ke-j lebih dari atau sama dengan waktu kendaraan ke=k untuk memulai pelayanan di konsumen ke-i ditambah waktu pelayanan konsumen ke-i dan ditambah waktu tempu perjalanan dari konsumen ke-i ke konsumen ke-j.

Jika ada lintasan dari i ke j dengan kendaraan k, maka:

$$T_{ik} + s_{ik} + Wt_{ij} \leq T_{jk}, \forall_{i,j} \in N, \forall k \in K$$

d Waktu kendaraan untuk memulai pelayanan di konsumen ke-i harus berada pada selang waktu $[a_i, b_i]$.

$$a_i \leq T_{ik} \leq b_i, \forall i \in N, \forall k \in K$$

e Setiap rute perjalanan berawal dari depot dan berakhir di depot.

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} X_{ijk} = 1, \forall i, j \in N, \forall k \in K$$

f Kekontinuan rute, artinya kendaraan yang mengunjungi setiap konsumen setelah selesai melayani akan meninggalkan konsumen tersebut.

$$\sum_{i=1}^{n} X_{ijk} - \sum_{i=1}^{n} X_{ijk} = 0, \forall i, j \in N, \forall k \in K$$

g Variabel keputusan X_{ijk} merupakan integer biner.

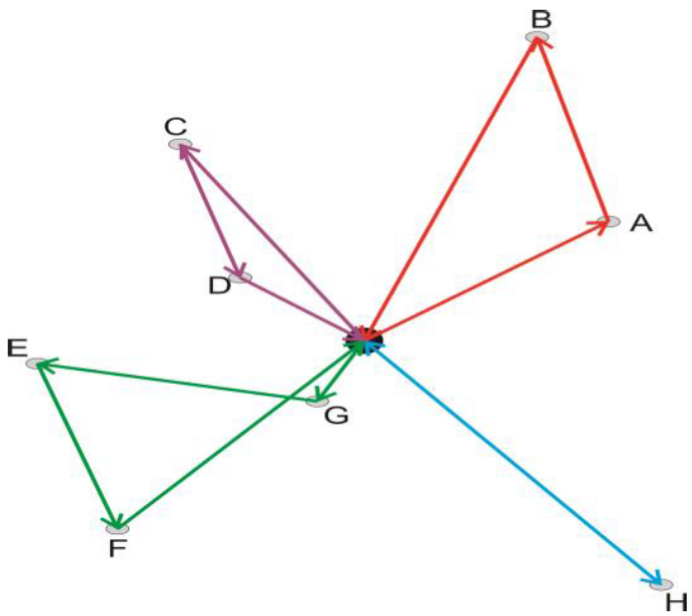
$$X_{ijk} \in \{0, 1\}, \forall i, j \in N, \forall k \in K$$

Formulasi model matematis CVRP tersebut pada intinya menekankan pada batasan *subtour elimination* yaitu mengeliminasi *subtour* supaya tidak terdapat subrute pada rute-rute yang terbentuk yang dikaitkan dengan batasan kapasitas kendaraan. Variabel keputusan hanya akan terdefinisi jika jumlah permintaan konsumen i dan konsumen j tidak melebihi kapasitas kendaraan.

2.5.3 Representasi Solusi

Solusi layak dari formulasi CVRP yang dihasilkan adalah himpunan rute kendaraan yang memiliki total waktu tempuh minimal dengan memenuhi semua kendala yang ada. Himpunan rute tersebut dapat dituliskan sebagai berikut:

$$R = \{R_1, R_2, ..., R_n\}$$



Gambar 2.11: Ilustrasi Solusi Layak CVRP

Solusi CVRP dapat digambarkan dalam bentuk graf yang setiap rute perjalanannya merupakan lintasan tertutup dengan depot sebagai simpul awal dan simpul akhir, sedangkan simpul lainnya adalah konsumen. Ilustrasi mengenai solusi layak dari CVRP seperti pada gambar 2.11.

Pada Gambar 2.11 terdapat delapan konsumen yaitu *A, B, C, D, E, F, G, H*. Dengan penggunaan salah satu algoritma eksak ataupun heuristik didapat sebuah solusi yang terdiri dari empat rute perjalanan kendaraan yang diawali dan diakhiri di depot serta memenuhi kendala yang ada pada CVRP. Rute pertama terdiri dari konsumen A dan B, rute kedua terdiri dari konsumen C dan D, rute ketiga terdiri dari konsumen E, F, dan G, dan rute terakhir hanya diisi oleh konsumen H saja. Dengan demikian, representasi solusi dari masalah CVRP tersebut adalah $R = \{0-A-B-0, 0-C-D-0, 0-E-F-G-0, 0-H-0\}$ dengan total waktu tempuh kendaraan $= Wt_{0A,t} + Wt_{AB,t} + Wt_{B0,t} + Wt_{0C,t} + Wt_{CD,t} + Wt_{D0,t} + Wt_{0E,t} + Wt_{EF,t} + Wt_{FG,t} + Wt_{G0,t} + Wt_{0H,t} + Wt_{H0,t}$

2.6 Algoritma *Floyd-Warshall*

2.6.1 Pengertian

Algoritma *Floyd-Warshall* diciptakan oleh R. Floyd pada tahun 1962. Algoritma *Floyd-Warshall* adalah salah satu pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu. Algoritma *Floyd-Warshall* ini akan memilih satu jalur terpendek dan teraman dari beberapa alternatif jalur yang telah dihasilkan dari proses kalkulasi.

Hal yang membedakan pencarian solusi menggunakan pemrograman dinamis dengan algoritma *greedy* adalah bahwa keputusan yang diambil pada tiap tahap pada algoritma *greedy* hanya berdasarkan pada informasi yang terbatas sehingga nilai optimum yang diperoleh pada saat itu Jadi pada algoritma *greedy*, kita tidak memikirkan konsekuensi yang akan terjadi seandainya kita memilih suatu keputusan pada suatu tahap. Dalam beberapa kasus, algoritma *greedy* gagal memberikan solusi terbaik karena kelemahan yang dimilikinya tadi. Di sinilah peran pemrograman dinamis yang mencoba untuk memberikan solusi yang memiliki pemikiran terhadap konsekuensi yang ditimbulkan dari pengambilan keputusan pada suatu tahap. Pemrograman dinamis mampu mengurangi pengenumerasian keputusan yang tidak mengarah ke solusi. Prinsip yang dipegang oleh pemrograman dinamis adalah prinsip optimalitas, yaitu jika solusi total optimal, maka bagian solusi sampai suatu tahap (misalnya tahap ke-i) juga optimal.

2.6.2 Karateristik Program Dinamis

Beberapa karakteristik yang dimiliki oleh program dinamis antara lain:

a Persoalan dibagi atas beberap tahap, yang setiap tahapnya hanya akan diambil satu keputusan.

b Masing-masing tahap terdiri atas sejumlah status yang saling berhubungan dengan status tersebut. Status yang dimaksud di sini adalah berbagai kemungkinan masukan yang ada pada tahap tersebut.

c Ketika masuk ke suatu tahap, hasil keputusan akan transformasi.

d Ongkos (beban) pada suatu tahap akan meningkat secara teratur seiring bertambahnya jumlah tahapan.

e Ongkos yang ada pada suatu tahap tergantung dari ongkos tahapan yang telah berjalan dan ongkos pada tahap itu sendiri.

f Keputusan terbaik pada suatu tahap bersifat independen terhadap keputusan pada tahap sebelumnya.

g Terdapat hubungan rekursif yang menyatakan bahwa keputusan terbaik dalam setiap status pada tahap k akan memberikan keputusan terbaik untuk setiap status pada tahap k+1

h Prinsip optimalitas berlaku pada persoalan yang dimaksud.

Dalam proses penyelesaian menggunakan program dinamis, pendekatan yang dilakukan bisa jadi ada dua macam, yaitu pendekatan maju (*forward*) dan mundur (*backward*), dan perlu untuk diketahui pula bahwa solusi yang dihasilkan dari kedua pendekatan itu adalah sama. Solusi dari program dinamis bisa jadi lebih dari satu macam.

2.6.3 *Pseudo-code* Algoritma *Floyd-Warshall*

Pseudo-code algoritma *Floyd-Warshall* adalah sebagai berikut

```
function fw(int[1..n,1..n] graph) {
    // Inisialisasi
    var int[1..n,1..n] jarak := graph
    var int[1..n,1..n] sebelum
    for i from 1 to n
            for j from 1 to n
                if jarak[i,j] < Tak-hingga
                    sebelum[i,j] := i
// Perulangan utama pada algoritma
for k from 1 to n
    for i from 1 to n
        for j from 1 to n
            if jarak[i,j] > jarak[i,k] + jarak[k,j]
                jarak[i,j] = jarak[i,k] + jarak[k,j]
                sebelum[i,j] = sebelum[k,j]
return jarak
}
```

Gambar 2.12: *Pseudo-code* dari algoritma *Floyd Warshall*

Gambar 2.12 adalah *pseudo-code* dari *Floyd-Warshall* yang merupakan langkah-langkah dari algoritma *Floyd-Warshall*. Langkah pertama dari algoritma *Floyd-Warshall* ialah dilakukan inisialisasi yang menjadi *verteks* asal dan *verteks* tujuan yang ditunjukan pada $var\ int[1..n, 1..n]\ jarak :=$ $graph$ dan $var\ int[1..n, 1..n]$ sebelum n dimana n adalah banyaknya *verteks* dalam graf. Algoritma *Floyd-Warshall* menggunakan sebuah fungsi pencarian jarak(i, j, k) untuk menemukan jalur terpendek dari i ke j dengan perantara simpul k. Dimana i dan j menunjukan baris dan kolom dari sebuah matriks jarak. Selanjutnya dari fungsi pencarian jarak(i, j, k) dapat dibangun sebuah iterasi (perulangan) sebagai berikut $if\ jarak\ [i, j] > jarak\ [i, k] + jarak\ [k, j]$. Perulangan ini merupakan inti dari algoritma *Floyd-Warshall*. Algoritma ini bekerja dengan menghitung jarak terpendek dari matriks (i, j) dengan dimulai $k = 1$ untuk semua pasangan matriks (i, j), kemudian hasil tersebut akan digunakan untuk menghitung matrik jarak terpendek dari fungsi jaraka$(i, j, 2)$ untuk semua pasangan (i, j) dan seterusnya. Algoritma *Floyd-Warshall* membandingkan semua kemungkinan lintasan pada graf untuk setiap *edge* dari semua *verteks*. Proses ini akan berlangsung hingga $k = n$ dan telah ditemukan jalur terpendek untuk semua pasagan (i, j) menggunakan simpul-simpul perantara. Sehingga algoritma *Floyd Warshall* memiliki *output* berupa titik jalur terpendek dari sebuah rute perjalanan ke titik tujuan.

Pada proses komputasinya, algoritma *Floyd-Warshall* mencari nilai optimal pada seluruh pasangan *verteks*. Pencarian jalur tercepat dengan mekanisme memperbarui data di dalamnya akan memberatkan kerja sistem jika menggunakan algoritma *Floyd- Warshall*, karena dibutuhkan proses komputasi ulang pada seluruh pasangan *verteks* setiap kali terjadi memperbarui data. Selanjutnya saat dilakukan proses pencarian, sistem hanya perlu menampilkan jalur tercepat sesuai dengan *verteks* awal dan *verteks* tujuan yang dimasukkan oleh *user* dari himpunan jalur tercepat yang telah didapatkan pada proses sebelumnya (Cormen et al. 2009).

Pada permasalahan *Capacitated Vehicle Routing Problem* (CVRP), Algoritma Floyd-Warshall digunakan untuk mencari waktu tempuh tercepat. Adapun langkah-langkah pemecahan masalah CVRP dengan Algoritma Floyd-Warshall adalah sebagai berikut:

a Langkah 1

Buat waktu tempuh antar-*verteks* ($Wt^{(0)}$) pada saat t.

b Langkah 2

$Wt = Wt^{(0)}$

c Langkah 3

Untuk $h = \{1, 2, 3, ..., n\}$, $i = \{1, 2, 3, ..., n\}$, $j = \{1, 2, 3, ..., n\}$

Jika $Wt_{ij,t} > Wt_{ih,t} + Wt_{hj,s},$ dengan $s = Wt_{ih,t}$

maka tukar $Wt_{ij,t}$ dengan $Wt_{ih,t} + Wt_{hj,s},$ dengan $s = Wt_{ih,t}$

d Langkah 4

$Wt^{*} = Wt$

e Langkah 5

Ulangi langkah 3 sampai didapatkan hasil yang minimum.

BAB III

PEMBAHASAN

3.1 Pemodelan Data Dari Graf

Pada tahap pemodelan graf, data yang ada diberikan beberapa perlakuan sehingga membentuk sebuah graf yang dibutuhkan oleh penelitian ini. Beberapa perlakuan tersebut adalah sebagai berikut:

a Satukan setiap *verteks* dari masing-masing rute yang telah dikumpulkan menjadi sebuah *connected graph* (graf terhubung).

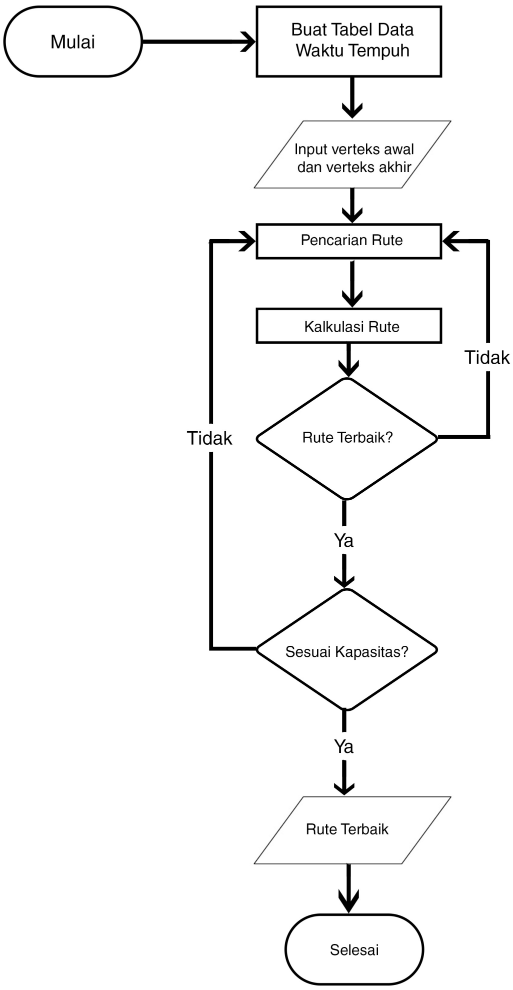
b Gunakan arah perjalanan pada rute sebagai aliran (*flow*) sehingga membentuk sebuah *directed graph* dari *connected graph* yang ada.

c Data berupa jarak tempuh yang didapatkan pada tahap pengumpulan data dan dijadikan sebagai bobot jarak. Implementasikan bobot-bobot tersebut sebagai aliran beban pada *directed graph* sehingga terbentuk sebuah *weighted graph* (graf berbobot).

Dari hasil perlakuan di atas, terbentuk sebuah graf terhubung yang memiliki aliran bobot *weighted graph* (graf berbobot). Bobot jarak adalah sebuah nilai atau aliran beban pada *weighted graph* yang mewakili panjang suatu *edge*, Satuan yang digunakan pada bobot jarak di penelitian ini adalah kilometer (km). Selanjutnya, graf dengan aliran bobot jarak akan diimplementasikan algoritma *Floyd-Warshall* untuk mendapatkan jalur terpendek.

3.2 Algoritma *Floyd-Warshall* Pada Model *Capacitated Vehicle Routing Problem*

Pada permasalahan *Capacitated Vehicle Routing Problem* (CVRP), Algoritma Floyd-Warshall digunakan untuk mencari waktu tempuh tercepat. Proses ini merupakan tahap-tahap untuk menyelesaikan permasalahan *Capacitated Vehicle Routing Problem* menggunakan algoritma *Floyd Warshall* sebagai waktu tempuh terpendek antar verteks, yang akan di sajikan dalam bentuk *Flowchart* dalam gambar 3.1.



Gambar 3.1: *Flowchart* dari algoritma *Floyd Warshall*

3.3 Deskripsi Dan Pemodelan Masalah *Capacitated Vehicle Routing Problem*

3.3.1 Data

Data yang digunakan dalam skripsi ini adalah data kegiatan distribusi barang dari JET, sebuah layanan jasa distribusi barang. Data yang diambil adalah waktu tempuh antara depot atau gudang tempat menampung barang dari satu region dengan kantor kantor agen perwakilan yang dibuat dalam bentuk data waktu tempuh antar lokasi, dan jumlah rata rata barang yang dikirim setiap harinya. Data jumlah rata rata barang yang dikirim setiap harinya diperoleh dari salah satu karyawan dari JET dan untuk data waktu tempuh dari depot ke agen perwakilan, diambil dari aplikasi peta *Google Maps*.

Tabel 3.1: Permintaan Data Tiap Agen

Verteks	Agen	Jarak (km)	Demand (Karung)
A	Gudang JET Kota Bekasi	0	0
B	Agen JET Villa Mutiara Gading 3	6.6	6
C	Agen JET Pondok Melati	3.9	6
D	Agen JET Harapan Baru	5.7	10
E	Agen JET Bintara Jaya	8.0	9
F	Agen JET Bulak Kapal	5.0	8
G	Agen JET Pekayon	5.1	10
H	Agen JET Rawalumbu	7.1	7
I	Agen JET Mustika Jaya	9.3	7
J	Agen JET Jatiwaringin	15.9	7
K	Agen JET Jatiasih	10.1	12
L	Agen JET Bantar Gebang	11.8	8
M	Agen JET Cibubur	25.6	22

3.3.2 Pendahuluan Model

JET adalah salah satu perusahaan yang bergerak pada layanan jasa pengiriman barang. Dalam suatu perusahaan jasa pengiriman barang, tentu-nya suatu rute distribusi yang optimal sangat dibutuhkan. Oleh karena itu, diperlukan suatu metode dalam penentuan rute yang efektif sehingga menghasilkan total waktu tempuh perjalanan yang optimal.

Dalam skripsi ini dibuat sebuah model graf yang menjadi dasar untuk melakukan perhitungan rute terpendek pada kasus pendistribusian barang oleh JET yang ada di regional Kota Bekasi. Model Graf dibuat berdasarkan pada jaringan jalan yang ada di lapangan. Data dasar yang digunakan untuk membuat model graf ini adalah jarak antar satu kantor agen JET dengan kantor agen JET yang lainnya.

3.3.3 Formulasi Masalah

Fungsi tujuan dari model penentuan rute kendaraan dalam pendistribusian barang pada skripsi ini adalah meminimumkan total waktu tempuh dari rute perjalanan kendaraan dengan memperhatikan batasan-batasan (kendala-kendala) yang ada sehingga rute-rute yang terbentuk merupakan rute-rute dengan waktu tempuh yang paling minimum yang memenuhi semua kendala kendala tersebut.

Adapun kendala-kendala yang dihadapi adalah:

a Setiap konsumen hanya dapat dikunjungi tepat satu kali oleh satu kendaraan.

b Total jumlah permintaan konsumen dalam satu rute tidak melebihi kapasitas kendaraan yang melayani rute tersebut.

c Setiap rute perjalanan kendaraan berawal dari depot.

d Setiap rute perjalanan kendaraan berakhir di depot.

e Kekontinuan rute, artinya setiap kendaraan yang mengunjungi suatu konsumen, setelah selesai melayani akan meninggalkan konsumen tersebut.

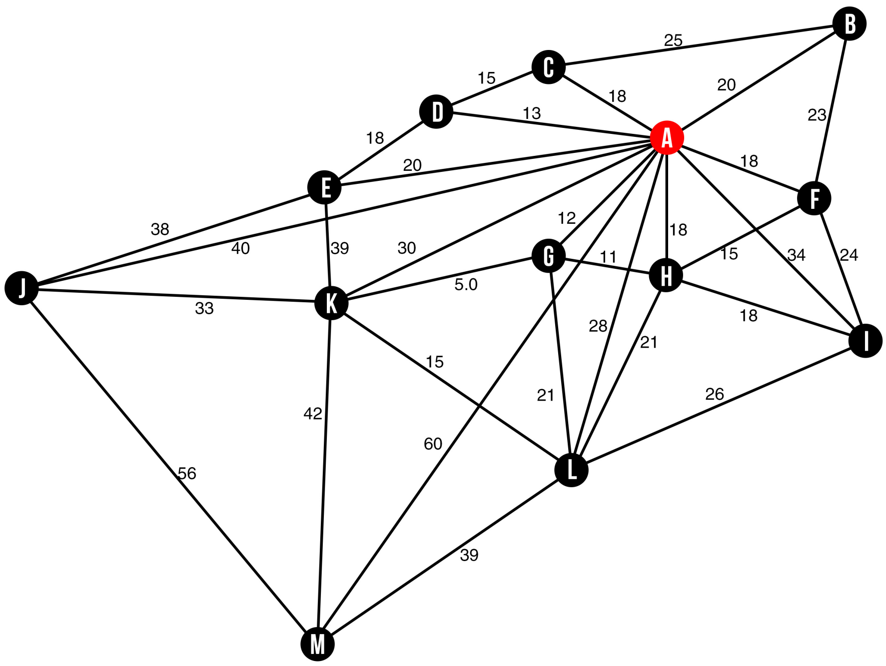
Untuk menyederhanakan masalah maka dalam Skripsi ini digunakan asumsi-asumsi sebagai berikut:

a Semua permintaan kantor agen dapat dipenuhi.

b Jumlah permintaan setiap agen sudah diketahui sebelumnya.

c Kendaraan yang digunakan mempunyai kapasitas yang sama yaitu 30 karung besar.

Model Graf yang dibuat pada skripsi ini memiliki tiga belas titik dan beberapa ruas jalan raya yang ada di Kota Bekasi. Model berupa *verteks* dan *edge* disertai bobot waktu tempuh antar *verteks* yang dihasilkan pada tahapan sebelumnya, pada tahap ini dibentuk menjadi beberapa graf. Bobot yang ada pada graf bernilai menit, dan untuk *verteks* yang tidak berhubungan satu sama lain akan dikenakan biaya tidak terhingga (∞) untuk bobotnya. Graf yang terbentuk dari perkumpulan rute jalan yang menghubungkan antar kantor agen JET dapat dilihat pada Gambar 3.2.



Gambar 3.2: Jaringan Jalur Distribusi Jasa Pengiriman Barang JET

Gambar di atas adalah pemetaan dari seluruh agen JET yang di kota Bekasi, dengan *verteks* A adalah gudang/depot yang akan mendistribusikan semua barang ke *verteks* lainnya. Berikut adalah representasi graf pada gambar 3.2:

Tabel 3.2: Waktu Tempuh Antar *Verteks*

	A	B	C	D	E	F	G	H	I	J	K	L	M
A	0	20	18	13	20	18	12	18	34	40	30	28	40
B	20	0	25	∞	∞	23	∞	∞	∞	∞	∞	∞	∞
C	18	25	0	15	∞	∞	∞	∞	∞	∞	∞	∞	∞
D	13	∞	15	0	18	∞	∞	∞	∞	∞	∞	∞	∞
E	20	∞	∞	18	0	∞	∞	∞	∞	38	39	∞	∞
F	18	23	∞	∞	∞	0	∞	15	24	∞	∞	∞	∞
G	12	∞	∞	∞	∞	∞	0	11	∞	∞	17	21	∞
H	18	∞	∞	∞	∞	15	11	0	18	∞	∞	21	∞
I	34	∞	∞	∞	∞	24	∞	18	0	∞	∞	26	∞
J	40	∞	∞	∞	38	∞	∞	∞	∞	0	33	∞	56
K	30	∞	∞	∞	39	∞	17	∞	∞	33	0	15	42
L	28	∞	∞	∞	∞	∞	21	21	26	∞	15	0	39
M	40	∞	∞	∞	∞	∞	∞	∞	∞	56	42	39	0

3.3.4 Pengolahan Model

Penyelesaian permasalahan CVRP untuk mendapatkan solusi rute distribusi pada data simulasi dilakukan dengan mengolah data yang telah diperoleh dengan menggunakan Algoritma Floyd-Warshall.

Langkah 1: Membentuk Semua Kemungkinan Rute Pertama

a Pada pembentukan rute pertama, akan dicari waktu tempuh terkecil dari gudang ke masing-masing konsumen.

   (a) A $\rightarrow$ B dengan waktu tempuh total 20 menit dan total demand 6 karung besar. Karena total demand 6 karung besar < kapasitas kendaraan, maka akan disisipkan kantor agen lain.

   (b) A $\rightarrow$ x $\rightarrow$ B, dengan $\forall x \in N - \{\text{A,B}\}$. Berikut adalah semua kemungkinan x.

Tabel 3.3: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh
A-F-B	14	41 Menit
A-C-B	12	43 Menit

Karena total demand kedua rute < kapasitas kendaraan, maka akan disisipkan konsumen lain.

   (c) A - y - x - B, dengan $\forall x \in N - \{\text{A,B}\}$, dan $\forall y \in N - \{\text{A}, x, \text{B}\}$ Berikut adalah semua kemungkinan y.

Tabel 3.4: Pembentukan Rute Pertama Pada Iterasi 2

Rute	Demand	Waktu Tempuh
A-I-F-B	21	81 Menit
A-H-F-B	21	56 Menit
A-D-C-B	22	53 Menit

Karena total demand ketiga rute < kapasitas kendaraan, maka akan disisipkan konsumen lain.

(d) A - z - y - x - B, dengan $\forall x \in N - \{\text{A,B}\}$, $\forall y \in N - \{\text{A}, x, \text{B}\}$, dan $\forall z \in N - \{\text{A}, x, y, \text{B}\}$ Berikut adalah semua kemungkinan z.

Tabel 3.5: Pembentukan Rute Pertama Pada Iterasi 3

Rute	Demand	Waktu Tempuh
A-H-I-F-B	28	83 Menit
A-I-H-F-B	28	90 Menit
A-G-H-F-B	31	61 Menit
A-L-H-F-B	29	87 Menit
A-E-D-C-B	31	78 Menit

Karena terdapat rute yang telah melebihi kapasitas kendaraan, maka stop proses. Diperoleh total waktu tempuh terkecil dari gudang JET Bekasi menuju agen JET Villa Mutiara Gading 3 Melalui Rute Gudang → Agen Rawalumbu → Agen Mustika Jaya → Agen Bulak Kapal → Agen Villa Mutiara Gading dengan total *demand* 28 Karung besar dan total waktu tempuh 83 menit.

(e) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Pondok Melati Melalui Rute Gudang → Agen Rawalumbu → Agen Bulak Kapal → Agen Villa Mutiara Gading 3 → Agen Pondok Melati dengan total *demand* 27 Karung besar dan total waktu tempuh 81 menit (Tabel 4.1, 4.2, dan 4.3 Lampiran).

(f) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Harapan Baru Melalui Rute Gudang → Agen Villa Mutiara Gading 3 → Agen Pondok Melati → Agen Harapan Baru dengan total *demand* 22 Karung besar dan total waktu tempuh 60 menit (Tabel 4.4, dan 4.5 Lampiran).

(g) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Bintara Jaya Melalui Rute Gudang → Agen Pondok Melati → Agen Harapan Baru → Agen Bintara Jaya dengan total *demand*

25 Karung besar dan total waktu tempuh 51 menit (Tabel 4.6, dan 4.7 Lampiran).

(h) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Bulak Kapal Melalui Rute Gudang → Agen Harapan Baru → Agen Pondok Melati → Agen Villa Mutiara Gading 3 → Agen Bulak Kapal dengan total *demand* 30 Karung besar dan total waktu tempuh 76 menit (Tabel 4.8, 4.9, dan 4.10 Lampiran).

(i) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Pekayon Melalui Rute Gudang → Agen Bulak Kapal → Agen Rawalumbu → Agen Pekayon dengan total *demand* 25 Karung besar dan total waktu tempuh 44 menit (Tabel 4.11, dan 4.12 Lampiran).

(j) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Rawalumbu Melalui Rute Gudang → Agen Bulak Kapal → Agen Mustika Jaya → Rawalumbu dengan total *demand* 22 Karung besar dan total waktu tempuh 54 menit (Tabel 4.13, dan 4.14 Lampiran).

(k) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Mustika Jaya Melalui Rute Gudang → Agen Pekayon → Agen Rawalumbu → Agen Mustika Jaya dengan total *demand* 24 Karung besar dan total waktu tempuh 41 menit (Tabel 4.15, dan 4.16 Lampiran).

(l) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Jatiwaringin Melalui Rute Gudang → Agen Pekayon → Agen Jatiasih → Agen Jatiwaringin dengan total *demand* 29 Karung besar dan total waktu tempuh 62 menit (Tabel 4.17, dan 4.18

Lampiran).

(m) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Jatiasih Melalui Rute Gudang → Agen Pekayon → Agen Jatiasih dengan total *demand* 22 Karung besar dan total waktu tempuh 29 menit (Tabel 4.19 Lampiran).

(n) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Bantar Gebang Melalui Rute Gudang → Agen Pekayon → Agen Agen Bantar Gebang dengan total *demand* 18 Karung besar dan total waktu tempuh 33 menit (Tabel 4.20 Lampiran).

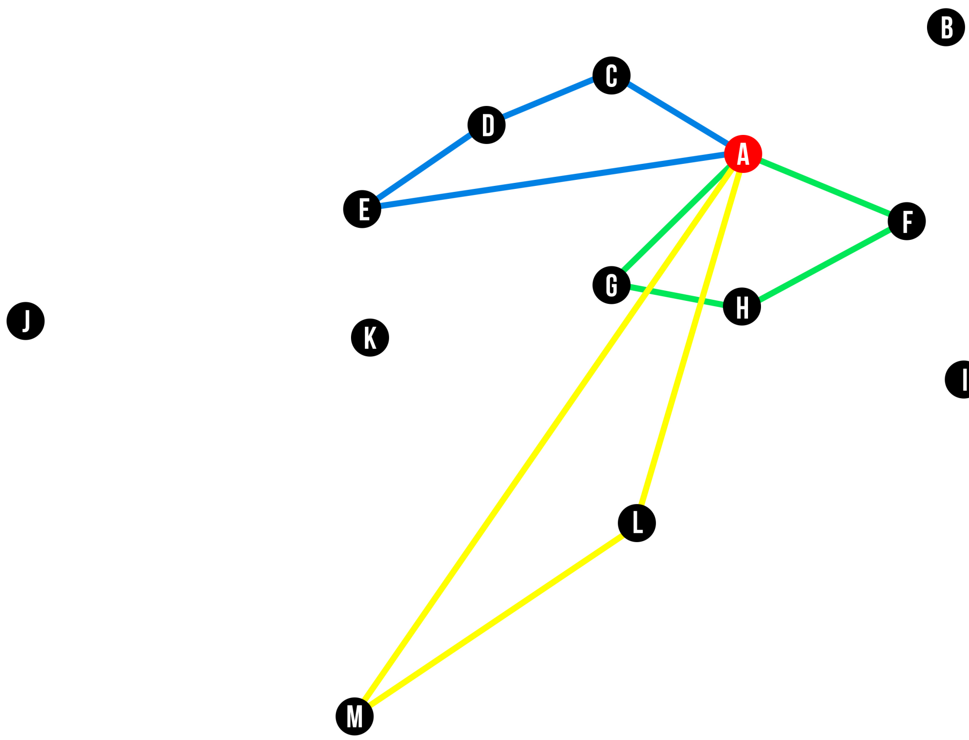
(o) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Cibubur Melalui Rute Gudang → Agen Bantar gebang → Agen Cibubur dengan total *demand* 30 Karung besar dan total waktu tempuh 67 menit (Tabel 4.21 Lampiran).

b Kemudian akan dicari rute dengan waktu tempuh terkecil dari depot sampai kembali ke depot. Berikut adalah semua kemungkinan rute:

Tabel 3.6: Total Rute Pertama Saat Kembali ke Depot

Rute	Demand (Karung)	Waktu Tempuh (Menit)
A-H-I-F-B-A	28	103
A-H-F-B-C-A	27	99
A-B-C-D-A	22	73
A-C-D-E-A	25	71
A-D-C-B-F-A	30	94
A-F-H-G-A	25	56
A-F-I-H-A	22	72
A-G-H-I-A	24	75
A-G-K-J-A	29	102
A-G-K-A	22	59
A-G-L-A	18	61
A-L-M-A	30	127

Dari semua kemungkinan rute diatas, didapat bahwa waktu tempuh tercepat dari gudang untuk kembali ke gudang adalah melalui Rute A → F → H → G → A atau dari gudang → agen Bulak Kapal → agen Rawalumbu → agen Pekayon → gudang dengan total waktu tempuh 56 menit dan total kapasitas muatannya sebanyak 25 karung besar, rute A → C → D → E → A atau dari gudang → agen Pondok Melati → agen Harapan Baru → agen Bintara Jaya → gudang dengan total waktu tempuh 71 menit dan total kapasitas muatannya sebanyak 25 karung besar, serta rute A → L → M → A atau dari gudang → agen Bantar Gebang → agen Cibubur → gudang dengan total waktu tempuh 127 menit dan total kapasitas muatannya sebanyak 30 karung besar.



Gambar 3.3: Pembentukan Rute Pertama, Kedua, dan Ketiga

c Jika terdapat agen yang belum dilayani, maka buat rute baru lagi.

Langkah 2: Membentuk Semua Kemungkinan Rute Selanjutnya

a Pada pembentukan rute keempat, akan dicari kembali waktu tempuh paling optimal dari gudang menuju agen yang belum dilayani oleh rute pertama, kedua dan ketiga.

(a) A B dengan waktu tempuh total 20 menit dan total demand 6 karung besar. Karena total demand 6 karung besar < kapasitas kendaraan, maka akan disisipkan kantor agen lain. Namun karena A - x - B dengan $\forall x \in N - \{\mathrm{A}, ,\mathrm{C}, \mathrm{D}, ..., \mathrm{H}, \mathrm{L}, \mathrm{M}\} = \infty$ (Karena *verteks* $\{\mathrm{A}, ,\mathrm{C}, \mathrm{D}, ..., \mathrm{H}, \mathrm{L}, \mathrm{M}\}$ telah dilayani pada rute sebelumnya), maka stop proses. Diperoleh rute dari gudang JET Bekasi menuju agen JET Villa Mutiara Gading 3 dengan total *demand* 6 Karung besar dan total waktu tempuh 20 menit.

(b) A I dengan waktu tempuh total 34 menit dan total demand 7 karung besar. Karena total demand 7 karung besar < kapasitas kendaraan, maka akan disisipkan kantor agen lain. Namun karena A - x - I dengan $\forall x \in N - \{\mathrm{A}, ,\mathrm{C}, \mathrm{D}, ..., \mathrm{H}, \mathrm{L}, \mathrm{M}\} = \infty$ (Karena *verteks* $\{\mathrm{A}, ,\mathrm{C}, \mathrm{D}, ..., \mathrm{H}, \mathrm{L}, \mathrm{M}\}$ telah dilayani pada rute sebelumnya), maka stop proses. Diperoleh rute dari gudang JET Bekasi menuju agen JET Mustika Jaya dengan total *demand* 7 Karung besar dan total waktu tempuh 34 menit.

(c) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Jatiwaringin Melalui Rute Gudang → Agen Jatiasih → Agen Jatiwaringin dengan total *demand* 19 Karung besar dan total waktu tempuh 63 menit (Tabel 4.22 Lampiran).

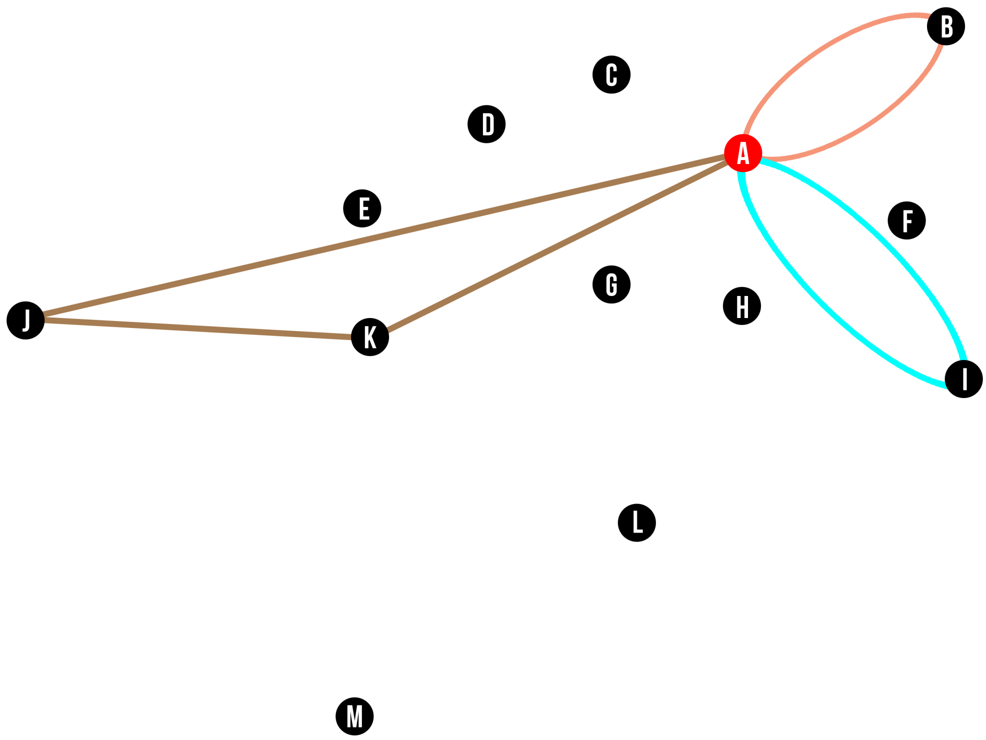
(d) Untuk total waktu tempuh terkecil dari gudang JET Bekasi menuju agen Jatiasih Melalui Rute Gudang $\rightarrow$ Agen Jatiwaringin $\rightarrow$ Agen Jatiasih dengan total *demand* 19 Karung besar dan total waktu tempuh 63 menit (Tabel 4.23 Lampiran).

b Kemudian akan dicari rute dengan waktu tempuh terkecil dari depot sampai kembali ke depot. Berikut adalah semua kemungkinan rute:

Tabel 3.7: Total Rute Ketiga Saat Kembali ke Depot

Rute	Demand (Karung)	Waktu Tempuh (Menit)
A-B-A	6	40
A-I-A	7	68
A-K-J-A	27	103
A-J-K-A	27	103

Dari semua kemungkinan rute diatas, didapat bahwa waktu tempuh terpendek dari gudang untuk kembali ke gudang adalah melalui Rute A $\rightarrow$ B $\rightarrow$ A atau dari gudang $\rightarrow$ agen Villa Mutiara Gading 3 $\rightarrow$ gudang dengan total waktu tempuh 40 menit dan total kapasitas muatannya sebanyak 6 karung besar, Rute A $\rightarrow$ I $\rightarrow$ A atau dari gudang $\rightarrow$ agen Mustika Jaya $\rightarrow$ gudang dengan total waktu tempuh 68 menit dan total kapasitas muatannya sebanyak 7 karung besar, serta rute A $\rightarrow$ K $\rightarrow$ J $\rightarrow$ A atau dari gudang $\rightarrow$ agen Jatiasih $\rightarrow$ agen Jatiwaringin $\rightarrow$ gudang dengan total waktu tempuh 103 menit dan total kapasitas muatannya sebanyak 27 karung besar.



Gambar 3.4: Pembentukan Rute Keempat, Kelima, dan Keenam

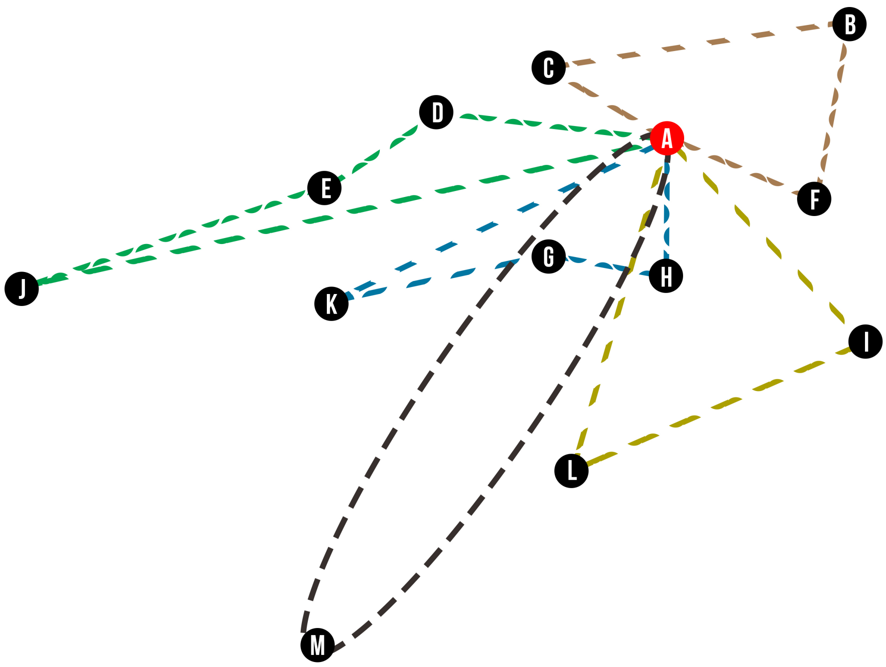
3.3.5 Perbandingan dengan Rute Saat Ini

Perbandingan rute hasil solusi tercepat dengan rute saat ini dilakukan dengan membandingkan rute kendaraan solusi optimal dengan rute kendaraan saat ini berdasarkan total waktu tempuh kendaraan. Hal tersebut dapat dilihat pada Tabel 3.9.

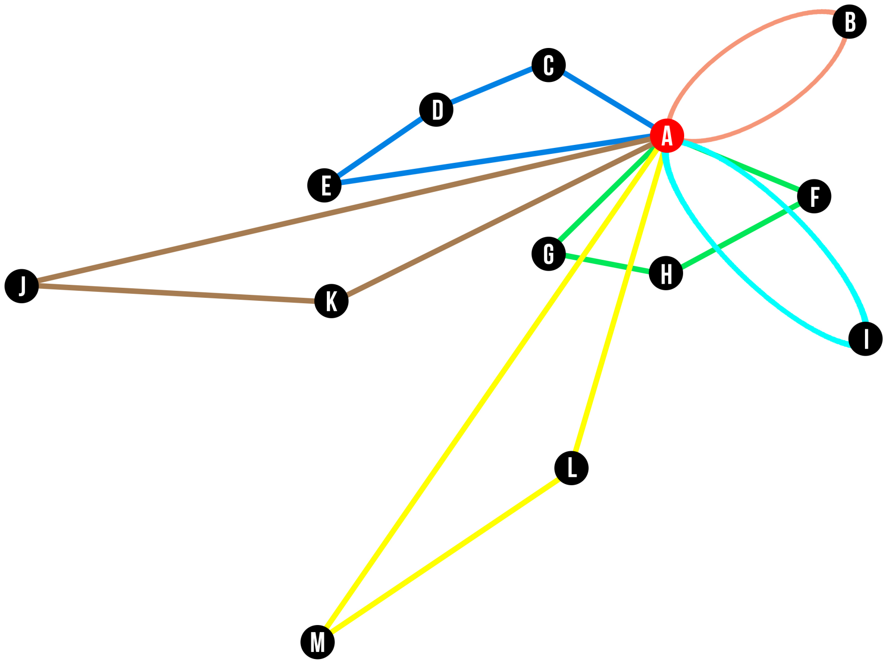
Tabel 3.8: Perbandingan rute solusi tercepat dan rute saat ini berdasarkan waktu tempuh kendaraan

	Rute Saat Ini (W_t / menit)	Solusi Optimal (W_t / menit)
	A-F-B-C-A (84) A-D-E-J-A (109) A-K-G-H-A (76) A-I-L-A (88) A-M-A (120)	A-C-D-E-A (71) A-F-G-H-A (56) A-L-M-A (127) A-B-A (40) A-I-A (68) A-K-J-A (103)
Total	477 Menit	465 Menit

Berdasarkan Tabel 3.9 dapat disimpulkan bahwa total waktu tempuh rute kendaraan yang dibuat memiliki total waktu tempuh yang lebih singkat dibandingkan dengan total waktu tempuh rute kendaraan saat ini. Pada rute solusi optimal, total waktu tempuh untuk kendaraan adalah 465 menit, sedangkan total waktu tempuh untuk kendaraan saat ini adalah 477 menit, sehingga ada penghematan waktu tempuh tempuh 12 menit. Perbedaan antara rute kendaraan solusi optimal dengan rute kendaraan saat ini dapat dilihat pada Gambar 3.5 dan 3.6.



Gambar 3.5: Rute Saat ini



Gambar 3.6: Rute Baru Dengan CVRP

Dari hasil perbandingan rute kendaraan solusi optimal dan rute kendaraan saat ini, perbedaan total waktu tempuh kendaraan dapat menjadi pertimbangan bagi perusahaan lainnya untuk menerapkan metode CVRP dengan algoritma *Floyd-Warshall* guna meminimalkan biaya distribusi dan mempersingkat waktu tempuh.

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan hasil pembahasan yang didapatkan pada skripsi ini, maka dapat dibuat beberapa simpulan:

1. Model *Capacitated Vehicle Routing Problem* dapat diselesaikan menggunakan Algoritma *Floyd-Warshall* dengan membentuk seluruh kemungkinan rute yang ditempuh sehingga menghasilkan waktu tempuh rute yang lebih cepat dibanding rute yang sudah diterapkan selama ini.

2. Dari hasil pembahasan, waktu tempuh rute kendaraan yang diselesaikan menggunakan Algoritma *Floyd-Warshall* lebih singkat dibandingkan dengan total waktu tempuh rute kendaraan saat ini. Pada rute yang diselesaikan oleh Algoritma *Floyd-Warshall*, total waktu tempuh untuk kendaraan adalah 465 menit, sedangkan total waktu tempuh untuk kendaraan saat ini adalah 477 menit, sehingga ada penghematan waktu tempuh tempuh hingga 12 menit.

4.2 Saran

1. Skripsi ini dapat dikembangkan untuk menyelesaikan masalah penentuan rute kendaraan dengan jumlah data yang lebih banyak.

2. Penelitian ini juga dapat dikembangkan untuk menyelesaikan masalah penentuan rute kendaraan dengan mempertimbangkan kendala kemacetan di jalan raya.

DAFTAR PUSTAKA

Bahri, Syamsul. 1997. *Penentuan Lintasan Terpendek Untuk Semua Pasangan Simpul*. Skripsi. Departemen Matematika. Institut Pertanian Bogor.

Budiarsyah, D.K. 2010. *Algoritma Dijkstra, Bellman-Ford, Dan Floyd-Warshall Untuk Mencari Rute Terpendek Dari Suatu Graf*. Bandung: Makalah Teknik Informatika Sekolah Tinggi Elektro dan Informatika Institut Teknologi Bandung.

Cormen T.H; Leiserson C.E; Rivest R.L; Stein C. 2009. *Introduction to Algorithms.Ed vol. 3*. Massachusetts (US): The MIT Press.

Donoriyanto, Dwi Sukma. 2012. *Analisis Kualitas Pelayanan Jasa Pengiriman Barang dengan metode Servqual dan QFD*. Jawa Timur: Jurnal Teknik Industri Universitas Pembangunan Nasional Veteran Jawa Timur.

Iskandar. 2010. *Model Optimasi Vehicle Routing Problem dan Implementasinya*. Tesis. Sekolah Pasca Sarjana. Institut Pertanian Bogor.

Jayanti, Ni Ketut D.A. 2014. *Penggunaan Algoritma Floyd-Warshall Dalam Masalah Jalur Terpendek Pada Penentuan Tata Letak Parkir*. Bali: Jurnal Informatika Sekolah Tinggi Ilmu Komputer Bali.

Munir, R. 2005. *Matematika Diskrit*. Bandung: Informatika.

Novandi, R.A. Diaz. 2007. *Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall dalam Penentuan Lintasan Terpendek (Single Pair Shortest Path)*. Makalah IF2251 Strategi Algoritmik, Bandung.

Prijono, Limbong. 2006. *Berhasil Gemilang Menguasai Matematika Diskrit Disertai Contoh Soal*. Bandung: Utomo.

Rosen, Kenneth H. 2011. *Discrete Mathematics and Its Applications,Seventh Edition*. New York: McGraw-Hill.

Utari, Resti Siti. 2013. *Penentuan Jalur Tercepat dan Terpendek Berdasarkan Kondisi Lalu Lintas di Kota Bogor Menggunakan Algoritme Dijkstra dan Algoritme Floyd-Warshall*. Skripsi. Departemen Ilmu Komputer. Institut Pertanian Bogor.

Zhan, F.B; Noon, C.E. 2000. *A Comparison Between Label-Setting and Label Correcting Algorithms for Computing One-to-One Shortest Path*, Journal of Geographic Information and Decision Analysis, Vol. 4 No. 2.

LAMPIRAN-LAMPIRAN

Membentuk Rute Pertama

Rute A - C (Gudang $\rightarrow$ Pondok Melati)

Tabel 4.1: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-B-C	12	45
A-D-C	16	28

Tabel 4.2: Pembentukan Rute Pertama Pada Iterasi 2

Rute	Demand	Waktu Tempuh (Menit)
A-F-B-C	20	66
A-E-D-C	25	51

Tabel 4.3: Pembentukan Rute Pertama Pada Iterasi 3

Rute	Demand	Waktu Tempuh (Menit)
A-H-F-B-C	27	81
A-I-F-B-C	27	106
A-J-E-D-C	32	111
A-K-E-D-C	37	102

Rute A - D (Gudang $\rightarrow$ Harapan Baru)

Tabel 4.4: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-C-D	16	33
A-E-D	19	38

Tabel 4.5: Pembentukan Rute Pertama Pada Iterasi 2

Rute	Demand	Waktu Tempuh (Menit)
A-B-C-D	22	60
A-J-E-D	26	96
A-K-E-D	31	87

Rute A - E (Gudang $\rightarrow$ Bintara Jaya)

Tabel 4.6: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-D-E	19	31
A-J-E	16	78
A-K-E	21	69

Tabel 4.7: Pembentukan Rute Pertama Pada Iterasi 2

Rute	Demand	Waktu Tempuh (Menit)
A-C-D-E	25	51
A-K-J-E	28	101
A-J-K-E	28	112
A-G-K-E	31	68
A-M-K-E	33	141
A-L-K-E	29	82

Rute A - F (Gudang $\rightarrow$ Bulak Kapal)

Tabel 4.8: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-B-F	14	43
A-I-F	15	58
A-H-F	15	33

Tabel 4.9: Pembentukan Rute Pertama Pada Iterasi 2

Rute	Demand	Waktu Tempuh (Menit)
A-C-B-F	20	66
A-H-I-F	22	60
A-L-I-F	23	78
A-G-H-F	25	38
A-L-H-F	23	64
A-I-H-F	22	67

Tabel 4.10: Pembentukan Rute Pertama Pada Iterasi 3

Rute	Demand	Waktu Tempuh (Menit)
A-D-C-B-F	30	76
A-G-H-I-F	32	65
A-L-H-I-F	30	91
A-M-L-I-F	45	149
A-K-L-I-F	35	95
A-G-L-I-F	33	83
A-H-L-I-F	30	89
A-L-G-H-F	33	75
A-K-G-H-F	37	73
A-K-L-H-F	35	81
A-G-L-H-F	33	69
A-I-L-H-F	30	96
A-M-L-H-F	45	135
A-L-I H-F	30	87

Rute A - G (Gudang → Pekayon)

Tabel 4.11: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-K-G	22	47
A-L-G	18	49
A-H-G	17	29

Tabel 4.12: Pembentukan Rute Pertama Pada Iterasi 2

Rute	Demand	Waktu Tempuh (Menit)
A-E-K-G	31	76
A-J-K-G	29	90
A-M-K-G	44	119
A-L-K-G	30	60
A-K-L-G	30	66
A-M-L-G	40	120
A-H-L-G	25	60
A-I-L-G	25	81
A-I-H-G	24	63
A-L-H-G	25	60
A-F-H-G	25	44

Rute A - H (Gudang → Rawalumbu)

Tabel 4.13: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-G-H	17	23
A-L-H	15	49
A-I-H	14	52
A-F-H	15	33

Tabel 4.14: Pembentukan Rute Pertama Pada Iterasi 2

Rute	Demand	Waktu Tempuh (Menit)
A-K-G-H	29	58
A-L-G-H	25	60
A-F-I-H	22	60
A-L-I-H	22	72
A-B-F-H	21	58
A-I-F-H	22	73
A-G-L-H	25	54
A-M-L-H	37	120
A-K-L-H	27	66
A-I-L-H	22	81

Rute A - I (Gudang → Mustika Jaya)

Tabel 4.15: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-F-I	15	42
A-H-I	14	36
A-L-I	15	54

Tabel 4.16: Pembentukan Rute Pertama Pada Iterasi 2

Rute	Demand	Waktu Tempuh (Menit)
A-H-F-I	22	57
A-B-F-I	21	67
A-F-H-I	22	51
A-G-H-I	24	41
A-L-H-I	22	67
A-H-L-I	22	65
A-G-L-I	25	59
A-M-L-I	37	125
A-K-L-I	27	71

Rute A - J (Gudang → Jatiwaringin)

Tabel 4.17: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-E-J	16	58
A-K-J	19	63
A-M-J	29	116

Tabel 4.18: Pembentukan Rute Pertama Pada Iterasi 2

Rute	Demand	Waktu Tempuh (Menit)
A-D-E-J	26	69
A-K-E-J	28	107
A-E-K-J	28	92
A-M-K-J	41	135
A-G-K-J	29	62
A-K-M-J	41	128
A-L-M-J	27	123
A-L-K-J	27	29.3

Rute A - K (Gudang → Jatiasih)

Tabel 4.19: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-E-K	21	59
A-J-K	19	73
A-G-K	22	29
A-M-K	34	102
A-L-K	20	43

Rute A - L (Gudang → Bantar Gebang)

Tabel 4.20: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-H-L	15	39
A-I-L	15	60
A-G-L	18	33
A-M-L	30	99
A-K-L	20	45

Rute A - M (Gudang → Cibubur)

Tabel 4.21: Pembentukan Rute Pertama Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-J-M	29	96
A-K-M	34	72
A-L-M	30	62

Membentuk Rute Keempat

Rute A - J (Gudang → Jatiwaringin)

Tabel 4.22: Pembentukan Rute Keempat Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-K-J	19	63

Rute A - K (Gudang $\rightarrow$ Jatiasih)

Tabel 4.23: Pembentukan Rute Keempat Pada Iterasi 1

Rute	Demand	Waktu Tempuh (Menit)
A-J-K	19	73

SURAT PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya yang bertanda tangan di bawah ini, mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta:

Nama	:	Sri Baskoro Bagus Pratikno
No. Registrasi	:	3125111213
Jurusan	:	Matematika
Program Studi	:	Matematika

Menyatakan bahwa skripsi ini yang saya buat dengan judul "**Implementasi Algoritma *Floyd-Warshall* Untuk Menentukan Rute Tercepat Pada Jalur Distribusi Jasa Pengiriman Barang**" adalah :

a Dibuat dan diselesaikan oleh saya sendiri.

b Bukan merupakan duplikat skripsi yang pernah dibuat oleh orang lain atau jiplakan karya tulis orang lain.

Pernyataan ini dibuat dengan sesungguhnya dan saya bersedia menanggung segala akibat yang timbul jika pernyataan saya tidak benar.

Jakarta, Agustus 2017

Yang membuat pernyataan

Sri Baskoro Bagus Pratikno

DAFTAR RIWAYAT HIDUP

SRI BASKORO BAGUS PRATIKNO. Lahir di Bekasi, 23 Agustus 1993. Anak kedua dari pasangan Bapak Sriyanto dan Ibu Sri Purwanti. Saat ini bertempat tinggal di Jalan Kintamani I RT:003/012 No.76, Bekasi 17115.

No. Ponsel	:	08567909393
Email	:	bagus807@gmail.com

Riwayat Pendidikan : Penulis mengawali pendidikan di TK Taman Indria selama dua tahun, dan kemudian melanjutkan pendidikan di SDN Sepanjang Jaya VIII kelurahan Sepanjang Jaya pada tahun 1999-2005. Setelah itu, penulis melanjutkan ke SMP Negeri 02 Bekasi hingga tahun 2008. Kemudian kembali melanjutkan ke SMA PGRI I Bekasi dan lulus tahun 2011. Di tahun yang sama penulis melanjutkan ke Universitas Negeri Jakarta (UNJ), jurusan Matematika, melalui jalur SNMPTN Tertulis.

Riwayat Organisasi : Selama di bangku perkuliahan, penulis aktif di berbagai organisasi kemahasiswaaan. Dalam tahun pertama, penulis mendapat kepercayaan sebagai staff Departemen Kewirausahaan BEMJ Matematika selama dua tahun. Di tahun yang sama penulis aktif diberbagai acara seputar pendidikan seperti Pelangi Matematika, Calculus Cup, Seminar Matematika, dan lain sebagainya.

Riwayat Pekerjaan : Penulis mulai menjadi pengajar matematika sejak tahun 2012. Pada tahun 2014, penulis bekerja di BRIngin Life Syariah. Pada tahun 2016, penulis memiliki Kedai Ayam Goreng Rawalumbu.