BAB II TINJAUAN PUSTAKA

A. Penelitian Terkait

Dalam penelitian ini ada beberapa penelitian yang relevan sebagai referensi. Berikut adalah penelitian yang pernah dilakukan oleh peneliti terdahulu yang dapat ditunjukkan menggunakan tabel 1 di bawah ini :

Tabel 1 Daftar penelitian terkait

No	Peneliti	Judul
1	Sulindawaty, Trinanda Syahputra (2015) [13]	**Judul :** Pendistribusian barang farmasi menggunakan Algoritma Djikstra (Studi Kasus : PT. AIR MAS CHEMICAL) **Metode :** Algoritma Prim dan Algoritma Djikstra **Software :** WpfGraph dan *Google Maps* **Hasil Penelitian :** Hasil yang diperoleh menggunakan Algoritma Djikstra yaitu sejauh 260 km dan perhitungan menggunakan Algoritma Prim yaitu sejauh 230,1 km.
2	Dicky Moriza, Hari Adiyanto, Yodi Nurdiansya (2016) [14]	**Judul :** Rute Pendistribusian Air Mineral dalam kemasan menggunakan Metode *Nearest Neighbour* dan *Branch and Bound* di PT. Agronesia Bmc* **Metode :** Metode *Nearest Neighbour* Dan *Branch And Bound* **Software :** *software LINGO* **Hasil Penelitian :** Total jarak tempuh untuk 27 pelanggan menggunakan metode nearest neighbour diperoleh sejauh 141,49 km dan total waktu tempuh = 18,51 jam, metode branch and bound diperoleh total jarak tempuh sejauh 135,39 km dan total waktu tempuh = 18,26 jam.
3	Muhammad Khoiruddin Harahap, Nurul Khairina (2017) [7]	**Judul :** Pencarian Jalur Terpendek dengan Algoritma Djikstra **Metode :** Algoritma Djikstra **Software : -** **Hasil Penelitian :** Penentuan jalur terpendek menggunakan Algoritma

		Dijkstra dimulai dari *vertex* awal ke *vertex* tujuan. Dimana masing-masing *vertex* memiliki nilai jarak yang telah ditetapkan.
4	Imaduddin Agil Firdaus, Indra Gita Anugrah (2019) [15]	**Judul :** Pemilihan Jalur Terpendek Dalam Pengiriman Bahan Bangunan Menggunakan Metode Djikstra **Metode :** Algoritma Djikstra **Software :** *Google Maps* **Hasil Penelitian :** Jarak yang dihasilkan dari gudang menuju pengiriman terakhir mendapatkan jarak berjumlah 23 km menggunakan kendaraan berupa Dump Truk. Dengan rute jalan II => V => IX => X => XVI.
5	Aldy Cantona, dkk (2020) [16]	**Judul :** Implementasi Algoritma Dijkstra Pada Pencarian Rute Terpendek ke Museum di Jakarta **Metode :** Algoritma Dijkstra **Software :** OS Ubuntu 18.04 LTS, Android Studio 3.5.2, 64 bit, Draw.io dan Framework flutter v1.9.1+hotfix.2. **Hasil Penelitian :** Penerapan Algoritma yang diterapkan pada smartphone cukup efektif, dari total 20 bobot menjadi 7 bobot. Dengan hal ini perlu 35% bobot untuk mencapai Museum Nasional.
6	Chandra Desparaja, R. Faris Gumelar, Nitta Fitria Anggraeni (2020) [17]	**Judul :** Pendistribusian Produk Kartu Seluler untuk Alternatif Rute Terpendek Menggunakan Metode Branch and Bound di PT. T **Metode :** Algoritma *Branch and Bound* **Software :** *Google Maps* **Hasil Penelitian : -** Rute terpendek dari rute 4 yaitu : *vertex* sumber – Distribution Center Cipadung Kidul – Distribution Center Taman Kopo Indah – Distribution Center Soreang – Distribution Center Ciparay – *vertex* sumber diperoleh 84,3 km. - Pada rute 10 yaitu : *vertex* sumber – Distribution Center Ciparay – Distribution Center Soreang – Distribution Center

		Taman Kopo Indah – Distribution Center Cipadung Kidul - *vertex* sumber diperoleh 84,3 km.
7	Hamdun Sulaiman, dkk (2020) [18]	**Judul :** Algoritma Dijkstra untuk Pendistribusian Carica Nida Food Wonosobo **Metode :** Algoritma Djikstra **Software : -** **Hasil Penelitian :** Pendistribusian Carica menggunakan Algoritma Djikstra dari *vertex* awal kembali lagi ke *vertex* awal. Rute untuk mendistribusikan produk carica adalah A → B → C → D → A dengan jarak tempuh 20,51 km.
8	Ikhsan Baharudin, Ahmad Jaka Purwanto, Teguh Rahayu Budiman, Muchammad Fauzi (2021) [19]	**Judul :** Implementasi Algoritma Djikstra untuk menentukan Jalur Terpendek dalam distribusi barang **Metode :** Algoritma Djikstra **Software :** *Google Maps* **Hasil Penelitian :** - jalur distribusi dari PT. X ke PT. Y melalui Polda Jawa Barat – Carefour dengan jarak tempuh 12,3 km - jalur distribusi dari PT. Y ke PT. Z melalui Simpang lima – Vie Hotel Westhoff dengan jarak tempuh = 10,7 km - total jarak tempuh dari PT. X ke PT. Y kemudian PT. Z adalah 23 km.
9	Khalisya Lintang, Resista Vikaliana (2021) [20]	**Judul** : Implementasi Flody Warshall Algorithm untuk Optimasi Distribusi J&T Express: Studi Kasus Pickup Distribution Center J&T Express Pasar Minggu **Metode** : Algoritma *Floyd Warshall* **Software** : *Google Maps* **Hasil Penelitian** : Rute optimum 1 diperoleh sejauh 300 meter, pada rute 2 sejauh 900 meter.
10	Anie Lusiani, dkk (2021) [21]	**Judul** : Algoritma Prim dalam Penentuan Lintasan Terpendek dan Lintasan Tercepat pada Pendistribusian Logistik Bulog Jawa Barat **Metode** : Algoritma Prim **Software** : *Google Maps*

		Hasil Penelitian : Optimalisasi Algoritma Prim diperoleh MST dengan jarak tempuh 192,6 km dan waktu tempuh 241 menit.

Pada penelitian ini, peneliti mengangkat topik penelitian dengan judul "Implementasi Algoritma Djikstra untuk Menentukan Rute Terpendek Distribusi Logistik Pemilu 2024 di Kecamatan Kesugihan". Berdasarkan uraian beberapa penelitian diatas, dapat disimpulkan bahwa perbedaan dan keterbaruan penelitian yang dilakukan oleh penulis dengan peneliti lainnya yaitu topik penelitian dan metode yang digunakan. Tujuan dari penelitian ini untuk merekomdasikan rute pada pendistribusian logistik pemilu yang akan datang serta menghitung biaya total pengeluaran selama proses distribusi. Hasil penelitian ini adalah sebuah rute yang dilalui untuk setiap Kantor Kepala Desa dengan jarak yang optimal. Sehingga mampu meminimalisir biaya pengeluaran dalam distribusi logistik.

B. Landasan Teori

1. Teori Dasar

Definisi 1. Graf G didefinisikan sebagai pasangan himpunan (V, E), ditulis dengan notasi $G = (V, E)$, dalam hal ini V adalah himpunan tidak kosong dari *vertex* dan E adalah kumpulan *edge* yang menghubungkan sepasang *vertex* [6].

Berikut definisi teori dasar graf menurut Chartrand (1978) [22], seperti definisi *orde*, *size*, $u - v\ walk$, $u - v\ trail$, $u - v\ path$, *cycle*, *adjacent*, dan *incident.*

Definisi 2. Dalam graf $G, V(G)$ adalah himpunan semua *vertex* pada graf dan elemen $E(G)$ adalah himpunan semua *edge* pada graf G. Banyaknya *vertex* pada graf G disebut *orde* graf G, dinotasikan $|V(G)|$. Banyaknya *edge* pada graf G disebut *size* G, dinotasikan dengan $|E(G)|$.

Definisi 3. *Vertex u* dan *v* pada graf di katakan *adjacent*, jika $uv \in E(G)$. Lebih lanjut, *vertex u* dikatakan *incident* dengan *edge e* dan *vertex v incident* dengan *edge e*, jika $e = uv \in E(G)$.

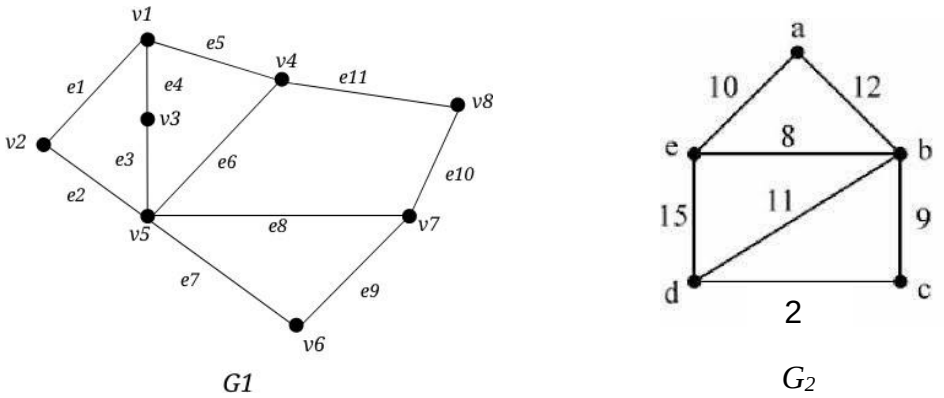
Definisi 4. Suatu $u - v\ walk$ dalam graf G adalah barisan bergantian antara *vertex* dan *edge* G yang dimulai dari *vertex u* dan berakhir di *vertex v*, dengan setiap *edge* menghubungkan *vertex* berurutan dalam barisan tersebut. Suatu $u - v\ trail$ pada graf G adalah $u - v\ walk$ dimana tidak ada pengulangan *edge* di dalamnya. Suatu $u - v\ path$ adalah $u - v\ walk$ yang tidak mengulang *vertex* manapun.

Definisi 5. *Circuit* adalah $u - v\ trail$ dimana $u = v$ yang memuat sedikitnya tiga *edge*. Suatu *circuit* dimana tidak ada pengulangan *vertex* di dalamnya (kecuali *vertex* awal dan akhir) disebut *cycle.*

Definisi 6. Graf G disebut terhubung jika setiap dua *vertex* dalam graf G dapat di tentukan suatu *path* yang menghubungkan dua *vertex* tersebut. Jika tidak, maka graf G tidak terhubung.

Definisi 7. Suatu graf G dikatakan graf sederhana jika graf G tidak memuat arah dan *loop*.

Definis 8. Graf G dikatakan berbobot apabila setiap *edge* nya diberi sebuah nilai (bobot).



Gambar 1 Graf G_1 dan G_2 Graf berbobot

Berdasarkan Gambar 1, graf G_1 merupakan contoh graf terhubung sederhana dengan $|V(G_1)| = 8$ dan $|E(G_1)| = 11$. Berdasarkan graf G_1 di diperoleh v_1 *adjacent* dengan v_2 dan e_1 *incident* dengan *vertex* v_1 dan v_2.

Berikut ini tabel 2 menjelaskan tentang contoh mengenai $u - v\ walk$, $u - v\ trail$, $u - v\ path$, $cycle$, dan $circuit$ berdasarkan gambar 1.

Tabel 2 Contoh rute pada *u – v walk*, *u - v trail*, *u – v path*, *cycle*, dan *circuit*

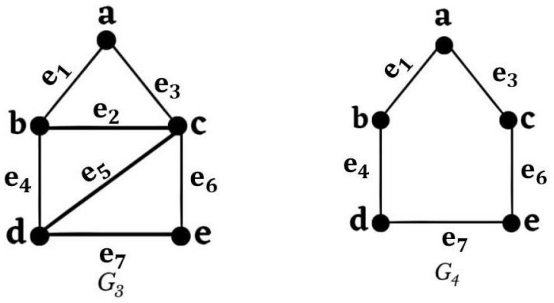
Contoh	Rute
$v_1 - v_6\ walk$	$v_1, e_1, v_2, e_2, v_5, e_7, v_6$
$v_1 - v_4\ trail$	$v_1, e_1, v_2, e_2, v_5, e_6, v_4$
$v_1 - v_8\ path$	$v_1, e_1, v_2, e_2, v_5, e_8, v_7, e_{10}, v_8$
cycle	$v_1, e_1, v_2, e_2, v_5, e_3, v_3, e_4, v_1$
circuit	$v_1, e_1, v_2, e_2, v_5, e_7, v_6, e_9, v_7, e_{10}, v_8, e_{11}, v_4, e_6, v_5, e_3, v_3, v_4, v_1,$

Pada graf G_2 adalah contoh graf berbobot dimana setiap *edge*nya mempunyai nilai.

2. Pohon Rentang Minimum (*Minimum Spanning Tree*)

Pada bagian ini dijelaskan terkait definisi dari sub graf, *spanning tree* dan graf pohon menurut Chartrand, Lesniak, & Zhang (2016) [23].

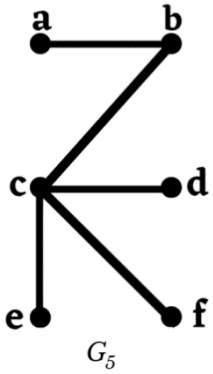
Definisi 9. Sebuah graf H disebut subgraf dari graf G jika $V(H) \subseteq V(G)$dan $E(H) \subseteq E(G)$, yang mana dapat ditulis $H \subseteq G$. Jika $V(H) = V(G)$ maka H adalah *spanning* subgraf dari G.



Gambar 2 Graf G_3 dan Graf G_4

Berdasarkan Gambar 2, graf G_4 merupakan *spanning subgraf* dari graf G_3 karena $V(G_3) = V(G_4)$.

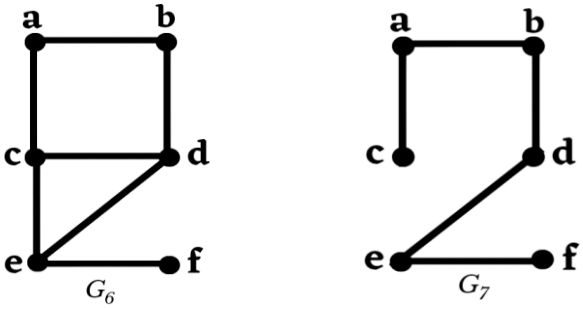
Definisi 10. Graf pohon adalah suatu graf terhubung dan tidak memiliki *circuit*.



Gambar 3 Graf G_5

Berdasarkan Gambar 3, graf G_5 merupakan graf pohon dengan 6 *vertex* dan 5 *edge* karena merupakan graf terhubung dan tidak memiliki *circuit*.

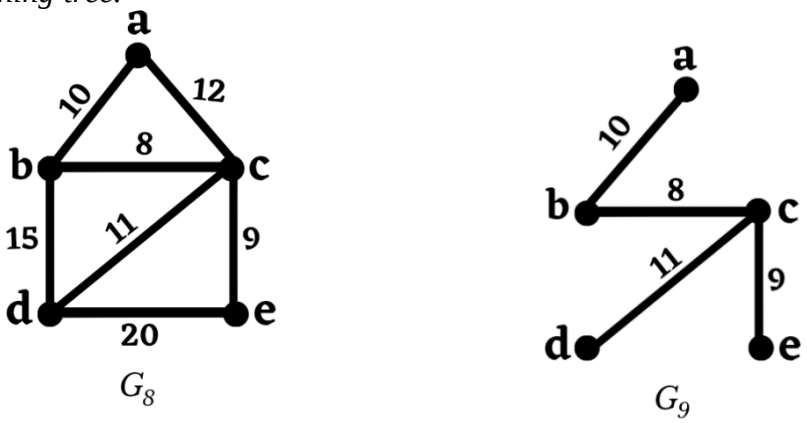
Definisi 11. Sebuah *Spanning tree* dari graf G adalah *spanning* subgraf dari G yang berupa graf pohon.



Gambar 4 Graf G_6 dan Graf G_7

Berdasarkan Gambar 4, graf G_7 merupakan *spanning tree* dari graf G_6.

Definisi 12. Sebuah *spanning tree* dari graf G yang bobotnya minimum diantara semua *spanning tree* dari graf G disebut dengan *minimum spanning tree*.



Gambar 5 Graf G_8 dan Graf G_9

Berdasarkan Gambar 5 graf G_9 merupakan *minimum spanning tree* darigraf G_8 karena mempunyai bobot terkecil yaitu 38.

3. Algoritma Djikstra

Algoritma Dijkstra ditemukan oleh Edsger Djikstra pada tahun 1959 [9], yang dapat digunakan untuk memecahkanpermasalahan jalur terpendek pada graf berarah atau graf tidak berarah dengan bobot *edge* yang tidak negatif [24].

Menurut Taha (2017) [25], berikut logistik yang akan digunakan pada langkah-langkah pelabelan *vertex* pada Algoritma Djikstra:

Keterangan :

u_i : bobot dari *vertex i* , dengan *i* = 1, 2, 3, 4, 5 ..
$u_{j,} i$: jarak terpendek yang bersifat permanen dari *vertex i* ke *vertex j*, dengan *j* = 2, 3, 4, 5 ..
d_{ij}, *i* : bobot antara *vertex i* ke *vertex j*

Pemberian label pada *vertex j* dari *vertex i* adalah:

$$[u_j, i] = [u_i + d_{ij}, i], d_{ij} \geq 0$$

Pada *vertex* v_1 labelnya akan ditetapkan sebagai $[0, -]$, yang menunjukan bahwa *vertex* awal tidak ada pendahulu. Label *vertex* dalam Algoritma Djikstra terdiri dari dua jenis, yaitu sementara dan permanen. Label permanen pada Algoritma Djikstra yaitu *vertex* dengan bobot terkecil dari hasil perhitungan menggunakan Algoritma Djikstra. Modifikasi label sementara apabila ditemukan rute yang lebih pendek ke suatu *vertex* dan jika tidak ada rute yang lebih pendek ditemukan, label sementara diubah menjadi label permanen. Berikut ini adalah proses dalam menentukan rute terpendek Algoritma Djikstra, yaitu :

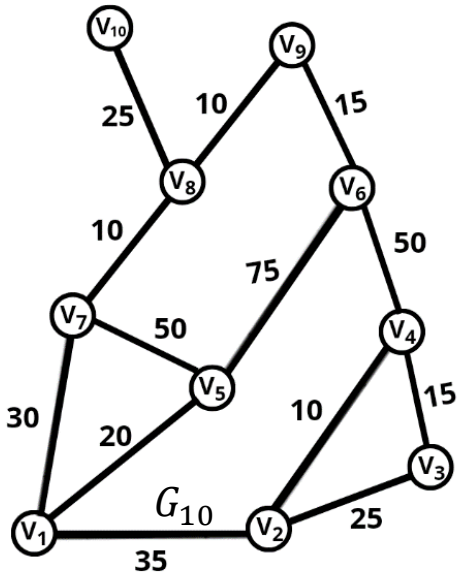
a. Beri label *vertex* awal v_1 dengan label permanen $[0, v_1]$, atur $i = 1$.
b. Hitung label sementara $[u_i + d_{ij}, i]$ untuk setiap *vertex j* yang *adjacent* dengan *vertex i* yang sudah berlabel permanen, dengan syarat *j* tidak berlabel permanen. Jika *vertex j* sudah dilabeli dengan $[u_j, k]$ melalui *vertex k* dan jika $u_i + d_{ij} < u_j$, ganti $[u_j, k]$ dengan $[u_i + d_{ij}, i]$.
c. Apabila semua *vertex* memiliki label permanen, maka proses berhenti. Namun, jika tidak, pilih label sementara $[ur, s]$ yang memiliki bobot terpendek dan mengubahnya menjadi label permanen. Jika terdapat lebih dari satu label dengan bobot yang sama, pilihlah salah satu secara kondisional dan mengubahnya menjadi label permanen. Langkah ini akan diulang kembali.

Berikut merupakan contoh penerapan Algoritma Djikstra. Diberikan data sebagai berikut:

Tabel 3 Tabel Jarak

V	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}
V_1	-	35	-	-	20	-	30	-	-	-
V_2	35	-	25	10	-	-	-	-	-	-
V_3	-	25	-	15	-	-	-	-	-	-
V_4	-	10	15	-	-	50	-	-	-	-
V_5	20	-	-	-	-	75	50	-	-	-
V_6	-	-	-	50	75	-	-	-	15	-
V_7	30	-	-	-	50	-	-	10	-	-
V_8	-	-	-	-	-	-	10	-	10	25
V_9	-	-	-	-	-	15	-	10	-	-
V_{10}	-	-	-	-	-	-	-	25	-	-

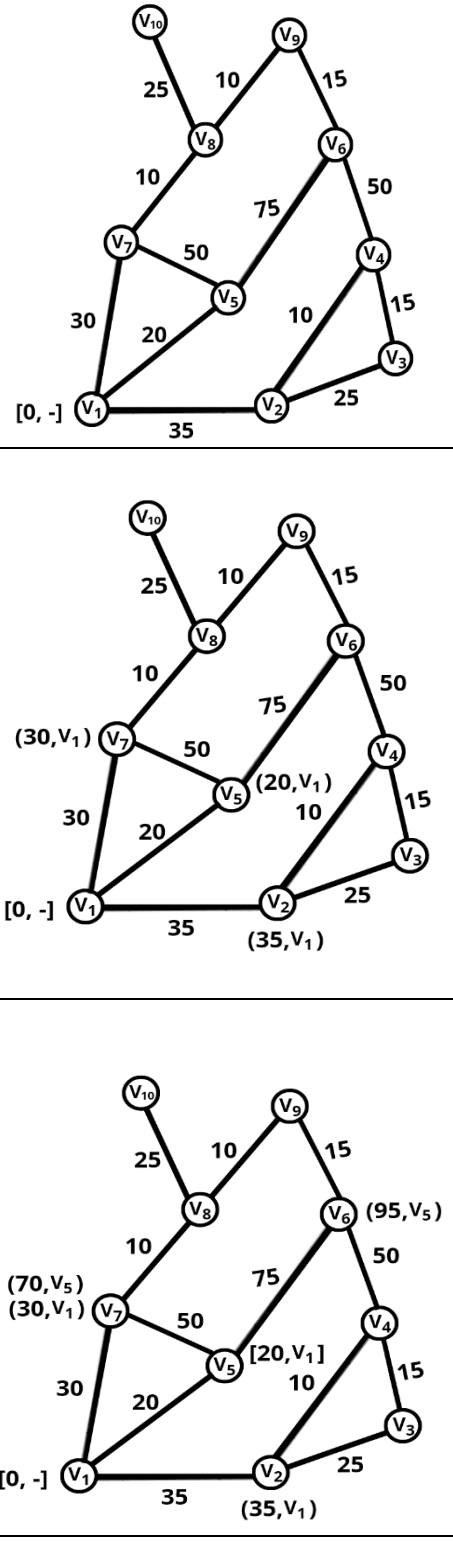
Tabel 3 merepresentasikan jarak antar V_1 sampai V_{10}. Selanjutnya diubah ke dalam bentuk graf sebagai berikut:

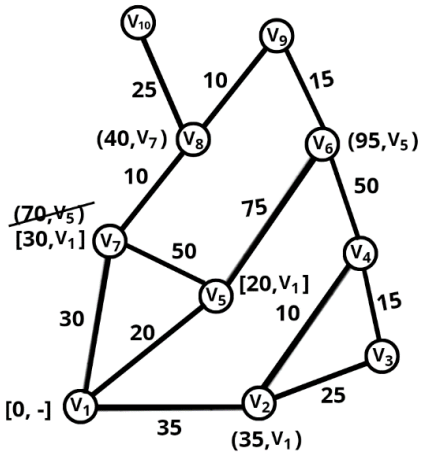
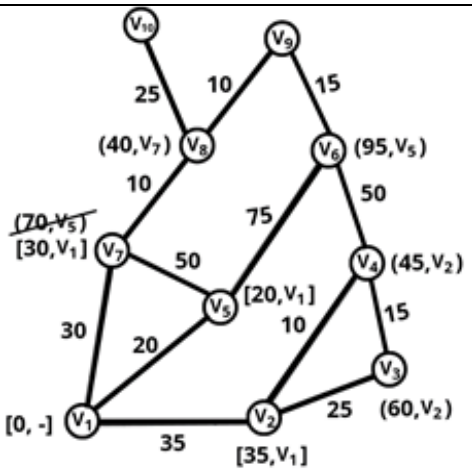


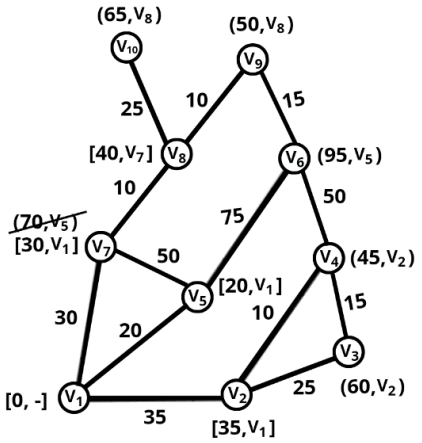
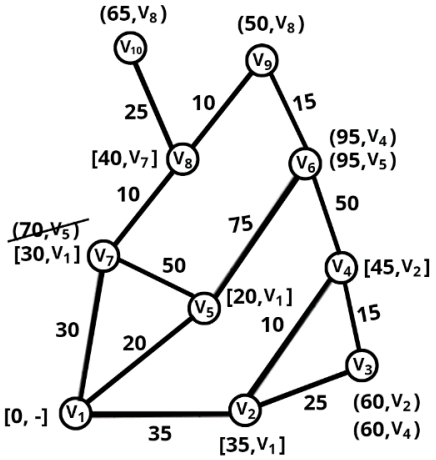
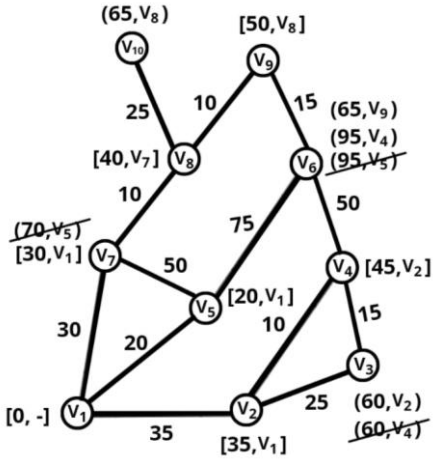
Gambar 6 Graf G_{10}

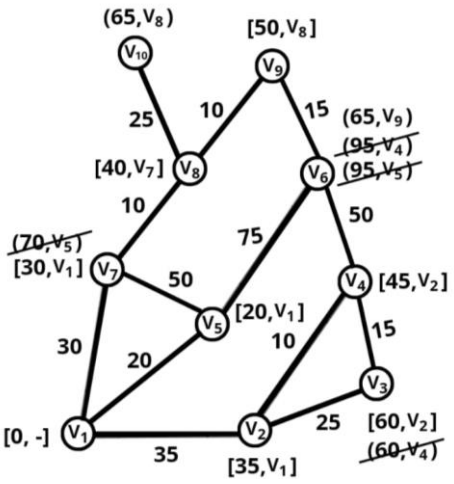
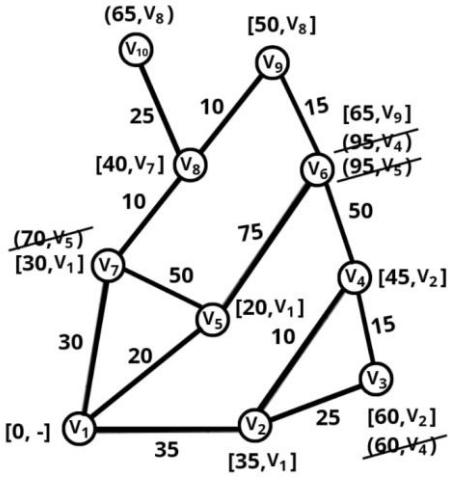
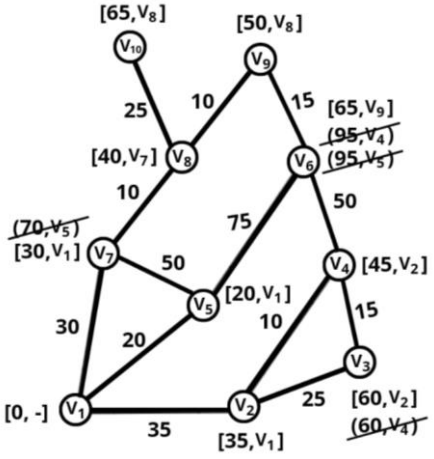
Gambar 6 merupakan representasi graf dari Tabel 3. Selanjutnya akan dicari rute terdekat dari V_1 ke semua *vertex* menggunakan Algoritma Djikstra pada tabel berikut ini.

Tabel 4 Tabel Perhitungan Algoritma Djikstra

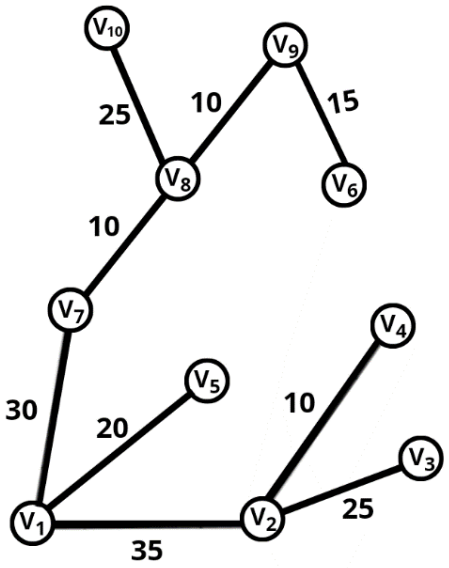
<table>
<tr><th>Iterasi</th><th>Vertex</th><th>Label</th><th>Graf</th></tr>
<tr><td>0</td><td>V_1</td><td>[0, −]Permanen</td><td rowspan="6"> 
</td></tr>
<tr><td rowspan="4">1</td><td>T : {V_1}</td><td>[0, −]</td></tr>
<tr><td>$V_1 - V_2$</td><td>(0 + 35, V_1) = (35, V_1)</td></tr>
<tr><td>$V_1 - V_5$</td><td>(0 + 20, V_1) = (20, V_1)</td></tr>
<tr><td>$V_1 - V_7$</td><td>(0 + 30, V_1) = (30, V_1)</td></tr>
<tr><td>Dipilih</td><td>$V_1 - V_5$</td><td>[20, V_1] Permanen</td></tr>
<tr><td rowspan="5">2</td><td>T : {V_1, V_5}</td><td>[20, V_1]</td><td rowspan="5">
</td></tr>
<tr><td>$V_1 - V_2$</td><td>(35, V_1)</td></tr>
<tr><td>$V_1 - V_7$
$V_5 - V_7$</td><td>(30, V_1) atau
(20 + 50, V_5) = (70, V_5)</td></tr>
<tr><td>$V_5 - V_6$</td><td>(20 + 75, V_5) = (95, V_5)</td></tr>
<tr><td>Dipilih</td><td>$V_1 - V_7$</td><td>[30, V_1] Permanen</td></tr>
</table>

Iterasi	*Vertex*	Label	Graf
3	T : $\{V_1, V_5, V_7\}$	[30, V_1]	
	$V_1 - V_2$	(35, V_1)	
	$V_5 - V_6$	(95, V_5)	
	$V_7 - V_8$	(30 + 10, V_7) = (40, V_7)	
Dipilih	$V_1 - V_2$	[35, V_1] Permanen	
4	T : $\{V_1, V_5, V_7, V_2\}$	[35, V_1]	
	$V_5 - V_6$	(95, V_5)	
	$V_7 - V_8$	(40, V_7)	
	$V_2 - V_3$	(35 + 25, V_2) = (60, V_2)	
	$V_2 - V_4$	(35 + 10, V_2) = (45, V_2)	
Dipilih	$V_7 - V_8$	[40, V_7]	

Iterasi	*Vertex*	Label	Graf
5	T : $\{V_1, V_5, V_7, V_2, V_8\}$	[40, V_7]	
	$V_5 - V_6$	(95, V_5)	
	$V_2 - V_3$	(60, V_2)	
	$V_2 - V_4$	(45, V_2)	
	$V_8 - V_9$	(40 + 10, V_8) = (50, V_8)	
	$V_8 - V_{10}$	(40 + 25, V_8) = (65, V_8)	
Dipilih	$V_2 - V_4$	[45, V_2]	
6	T : $\{V_1, V_5, V_7, V_2, V_8, V_4\}$	[45, V_2]	
	$V_5 - V_6$ $V_4 - V_6$	(95, V_5) atau (45 + 50, V_4) = (95, V_4)	
	$V_2 - V_3$ $V_4 - V_3$	(60, V_2) atau (45 + 15, V_4) = (60, V_4)	
	$V_8 - V_9$	(50, V_8)	
	$V_8 - V_{10}$	(65, V_8)	
Dipilih	$V_8 - V_9$	[50, V_8]	
7	T : $\{V_1, V_5, V_7, V_2, V_8, V_4, V_9\}$	[50, V_8]	
	$V_4 - V_6$ $V_9 - V_6$	(95, V_4) atau (50 + 15, V_9) = (65, V_9)	
	$V_2 - V_3$	(60, V_2)	
	$V_8 - V_{10}$	(65, V_8)	
Dipilih	$V_2 - V_3$	[60, V_2]	

Iterasi	*Vertex*	Label	Graf
8	T : $\{V_1, V_5, V_7, V_2, V_8, V_4, V_9, V_3\}$	[60, V_2]	
	$V_9 - V_6$	(65, V_9)	
	$V_8 - V_{10}$	(65, V_8)	
Dipilih	$V_9 - V_6$	[65, V_9]	
9	T : $\{V_1, V_5, V_7, V_2, V_8, V_4, V_9, V_3, V_6\}$	[65, V_9]	
	$V_8 - V_{10}$	(65, V_8)	
Dipilih	$V_8 - V_{10}$	[65, V_8]	
10	T : $\{V_1, V_5, V_7, V_2, V_8, V_4, V_9, V_3, V_6, V_{10}\}$	[65, V_8]	

Dari hasil tabel diatas maka graf yang dihasilkan adalah :



Gambar 7 Hasil akhir model graf Algoritma Djikstra

Karena semua *vertex* sudah terpilih, maka Jarak terpendeknya adalah:

Tabel 5. Hasil Rute Terpendek

Vertex	*Vertex* tujuan	*Vertex* yang dilalui	Jarak (Meter)
-	V_1	-	0
V_1	V_5	$V_1 \rightarrow V_5$	20
V_1	V_7	$V_1 \rightarrow V_7$	30
V_1	V_2	$V_1 \rightarrow V_2$	35
V_1	V_8	$V_1 \rightarrow V_7 \rightarrow V_8$	40
V_1	V_4	$V_1 \rightarrow V_2 \rightarrow V_4$	45
V_1	V_9	$V_1 \rightarrow V_7 \rightarrow V_8 \rightarrow V_9$	50
V_1	V_3	$V_1 \rightarrow V_2 \rightarrow V_3$	60
V_1	V_6	$V_1 \rightarrow V_7 \rightarrow V_8 \rightarrow V_9 \rightarrow V_6$	65
V_1	V_{10}	$V_1 \rightarrow V_7 \rightarrow V_8 \rightarrow V_{10}$	65

4. Logistik Pemilu

Logistik adalah suatu proses perencanaan, pelaksanaan, dan pengendalian aliran barang atau jasa dari *vertex* asal ke *vertex* konsumen dengan biaya yang efektif. Distribusi logistik pemilu yaitu proses Penyaluran barang-barang logistik dari KPU Provinsi ke KPU di bawahnya, serta dari KPU Kabupaten/Kota ke PPK, PPS, hingga TPS, dengan menggunakan alat transportasi. Pada konteks ini, logistik berkaitan dengan pengiriman dan distribusi bahan dan perlengkapan yang dibutuhkan untuk keperluan Pemilihan Umum 2024 di Kecamatan Kesugihan diantaranya kotak suara surat suara, tinta, bilik pemungutan suara, segel, dan alat untuk mencoblos pilihan, serta alat bantu tunanetra dan dukungan perlengkapan lainnya [26].

Oleh karena itu, Logistik Pemilu 2024 perlu dikelola dengan baik dalam segala aspek, mulai dari perencanaan kebutuhan dan penganggarannya, pengadaan, pendistribusian, hingga pemeliharaan dan inventarisasi.

5. POM-QM

Program *Operations Management – Quantitative Methods (POM-QM)* diperkenalkan sebagai aplikasi untuk memudahkan dan meminimalisir terjadinya *human error* pada perhitungan secara manual. *Software* ini awalnya dirilis dalam bentuk DOS pada tahun 1989 dengan nama PC-POM. Kemudian, versi pertama untuk Windows, yaitu *QM for Windows* (versi 1.0), disebarkan pada musim panas 1996 oleh Howard J. Weiss untuk membantu menyusun prakiraan dan anggaran untuk produksi bahan baku menjadi produk jadi atau setengah jadi dalam proses pabrikasi. *POM-QM* adalah sebuah program komputer yang digunakan untuk memecahkan masalah dalam praktik manajemen operasi untuk mencapai optimalisasi dalam berbagai aspek bisnis dan operasi yang bersifat kuantitatif [27].

Modul-modul yang terdapat di dalam *software* ini sudah sudah berkembang lebih pesat dan dapat digunakan untuk menghitung berbagai permasalahan. Pada penelitian ini, modul yang akan digunakan yaitu modul Network untuk mencari rute terpendek (*shortes path*) dengan versi *software* QM for Windows V5.

6. Biaya Logistik

Biaya transportasi rute kendaraan dari Kantor Kecamatan Kesugihan ke masing-masing Kantor Kepala Desa dengan rumus sebagai berikut [28] :

$$Biaya\ Bahan\ Bakar = \frac{\sum_{i=1}^{4} 2\ \times L_i}{jarak\ tempuh\ perliter} \times\ varibel\ cost$$

Keterangan :

i : Kantor Kepala Desa dengan *i = 1, 2, 3, 4*

L_i : Total jarak tiap Kantor Kepala Desa

$variabel\ cost$: Biaya solar perliter

Adapun untuk rumus Biaya Total sebagai berikut:

$$Biaya\ Total = ST + BBB + HS + KS$$

Keterangan :

ST : Sewa Truk

BBB : Biaya Bahan Bakar

HS : Honor Sopir

KS : Konsumsi Sopir