

Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra

Sri Andayani¹, Endah Wulan Perwitasari²

¹Jurusan Teknik Informatika, Sekolah Tinggi Teknik Musi, Palembang 30113
E-mail : andayani_s@sttmusi.ac.id

²Jurusan Teknik Informatika Universitas Musamus, Merauke 99612
E-mail : dek_endah@yahoo.com

ABSTRAK

Sampah adalah sisa kegiatan sehari-hari manusia dan proses alam yang berbentuk padat. Setiap sampah pasti menghasilkan sampah dalam jumlah yang variatif setiap harinya. Jumlah timbunan sampah semakin meningkat seiring dengan peningkatan pertumbuhan penduduk kota. Peningkatan timbunan sampah merupakan konsekuensi dari peningkatan kualitas dan perubahan pola hidup masyarakat. Oleh karena itu, laju timbunan sampah harus diikuti oleh peningkatan kualitas pengelolaan sampah untuk menghindari adanya dampak negatif yang timbul dari keberadaan sampah. Peningkatan pengelolaan sampah dapat dilakukan dengan pendistribusian sampah yang terkoordinasi dengan baik meliputi rute kendaraan, kendaraan sampai dengan minimasi ongkos distribusi dengan armada yang tersedia. Pemilihan rute kendaraan merupakan hal yang penting untuk menentukan total jarak terpendek perjalanan armada. Untuk dapat menentukan rute terpendek maka dibangun suatu aplikasi yang menggunakan algoritma Dijkstra. Algoritma Dijkstra memilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Algoritma Dijkstra membutuhkan parameter tempat asal dan tempat tujuan sehingga menghasilkan jarak terpendek dari tempat asal ke tempat tujuan beserta rutenya.

Kata kunci : Sampah, Dijkstra, rute, jarak

1. PENDAHULUAN

Sampah adalah sisa kegiatan sehari-hari manusia dan proses alam yang berbentuk padat. Setiap individu pasti menghasilkan sampah dalam jumlah yang variatif setiap harinya. Jumlah timbunan sampah semakin meningkat seiring dengan peningkatan pertumbuhan penduduk kota. Peningkatan timbunan sampah merupakan konsekuensi dari peningkatan kualitas dan perubahan pola hidup masyarakat. Oleh karena itu, laju timbunan sampah harus diikuti oleh peningkatan kualitas pengelolaan sampah untuk menghindari adanya dampak negatif yang mungkin ditimbulkan dari keberadaan sampah [7]

Pendistribusian sampah yang terkoordinasi dengan baik termasuk salah satu usaha peningkatan kualitas pengelolaan sampah. Pendistribusian sampah merupakan salah satu bagian penting dari kegiatan distribusi barang atau jasa yang dilakukan instansi pemerintah ataupun perusahaan tertentu.

Permasalahan distribusi sampah melibatkan beberapa pertimbangan utama meliputi rute kendaraan, kendaraan sampai dengan minimasi ongkos distribusi, sehingga dapat memperluas wilayah pelayanan dari pengambilan sampah dengan armada yang terbatas [3]. Masalah yang berkaitan dengan pendistribusian sampah diantaranya membuat keputusan-keputusan mengenai rute pengambilan sampah. Pemilihan rute kendaraan akan menentukan total jarak perjalanan armada.

Karakteristik permasalahan penentuan rute pengambilan sampah yaitu terdapat depo dimana kendaraan berangkat dan pulang, tiap konsumen tepat dilayani satu kali dalam sebuah rute, kapasitas yang diangkut dalam setiap rute tidak lebih dari kapasitas maksimal kendaraan pengangkut [5]. Sehingga rute yang optimal adalah rute yang memenuhi karakteristik permasalahan penentuan rute pengambilan sampah.

2. TINJAUAN PUSTAKA

Teori Graf

Sebuah graf G didefinisikan sebagai pasangan himpunan (V,E) di mana V =himpunan verteks $\{v_1,v_2,\dots,v_n\}$ dan E =himpunan edge(arc) yang menghubungkan verteks-verteks $\{e_1,e_2,\dots,e_n\}$ atau dapat ditulis dengan notasi $G=(V,E)$ Setiap sisi berhubungan dengan satu atau dua simpul. Dua buah simpul dikatakan berhubungan atau bertangga (adjacent) jika ada sisi yang menghubungkan keduanya. Berdasarkan orientasi yang ada pada sisinya, graf dapat dikelompokkan menjadi dua yaitu Graf berarah (direct graf) yaitu graf yang setiap sisinya diberikan arah sehingga untuk dua simpul v_i dan v_j , maka $(v_i,v_j) \neq (v_j,v_i)$ dan graf tak berarah (undirect graf) yaitu graf yang sisinya tidak mengandung arah sehingga untuk dua simpul v_i dan v_j maka $(v_i,v_j) = (v_j,v_i)$. Selain itu juga dikenal graf berbobot yaitu graf yang sisinya memiliki bobot atau nilai [1]

Path Terpendek

Path terpendek adalah suatu persoalan untuk mencari lintasan antara dua atau lebih simpul pada graf berbobot yang gabungan bobot sisi graf yang dilalui berjumlah paling minimum. Pada graf berbobot terdapat optimasi yang dapat dinyatakan dalam jarak antar kota, waktu pengiriman pesan, biaya dan sebagainya. Dalam hal ini bobot harus bernilai positif, walau dalam hal lain dapat bernilai negatif. Lintasan terpendek dengan verteks awal s dan verteks tujuan t didefinisikan sebagai lintasan terpendek dari s dan t dengan bobot minimum dan berupa lintasan sederhana (simple path).

Algoritma Dijkstra

Algoritma Dijkstra diterapkan untuk mencari lintasan terpendek pada graf berarah. Namun, algoritma ini juga benar untuk graf tak berarah. Algoritma Dijkstra mencari lintasan terpendek dalam sejumlah langkah. Algoritma ini menggunakan prinsip greedy. Prinsip greedy pada algoritma dijkstra menyatakan bahwa pada setiap langkah kita memilih sisi yang berbobot minimum dan memasukannya dalam himpunan solusi.

Contoh penerapan algoritma dijkstra adalah lintasan terpendek yang menghubungkan antara dua kota berlainan tertentu (*Single-source Single-destination Shortest Path Problems*). Cara kerja algoritma Dijkstra memakai strategi greedy, di mana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Algoritma Dijkstra membutuhkan parameter tempat asal, dan tempat tujuan. Hasil akhir dari algoritma ini adalah jarak terpendek dari tempat asal ke tempat tujuan beserta rutenya.

Misalkan sebuah graf berbobot dengan n buah simpul dinyatakan dengan matriks $M=[m_{ij}]$, yang dalam hal ini:

$$m_{ij} = \text{bobot sisi } (i,j) \text{ (pada graf tak berarah } m_{ij} = m_{ji})$$

$$m_{ii} = 0$$

$$m_{ij} = \infty, \text{ jika tidak ada sisi dari simpul } i \text{ ke simpul } j$$

Selain matriks M , juga menggunakan tabel $S=[s_i]$, yang dalam hal ini:

$$s_i = 1, \text{ jika simpul } i \text{ termasuk ke dalam lintasan terpendek}$$

$$s_i = 0, \text{ jika simpul } i \text{ tidak termasuk ke dalam lintasan terpendek}$$

Dan tabel $D=[d_i]$, yang dalam hal ini

$$d_i = \text{panjang lintasan dari simpul awal } a \text{ ke simpul } i$$

Penelitian Terdahulu

Universitas Sam Ratulangi Manado adalah salah satu perguruan tinggi di Sulawesi Utara yang terdiri atas 11 fakultas dan satu gedung rektorat. Setiap fakultas dan rektorat terhubung dengan fasilitas jalan raya. Secara matematis kondisi ini dapat direpresentasikan sebagai sebuah graf yang bisa diterapkan untuk mencari lintasan terpendek. Pada penelitian ini akan dicari lintasan terpendek dari FMIPA ke rektorat dan fakultas lainnya. Dengan menggunakan algoritma Dijkstra, lintasan terpendek dari FMIPA diperoleh dengan memilih minimum lokal atau akses dengan jarak terdekat dari setiap lokasi yang kemudian digabungkan menjadi sebuah kumpulan lintasan dari satu lokasi ke lokasi lainnya dengan jarak terpendek [9]

Pencarian rute terpendek merupakan salah satu persoalan dalam teori graf dan algoritma Dijkstra dengan mengembangkan sistem pencarian rute terpendek tempat wisata di Bali. Sistem yang dihasilkan berupa sistem informasi geografis yang berbentuk web dengan script pHP dan MySQL sebagai pengelola basis datanya [4].

Perlindungan masyarakat dari resiko ancaman bencana dapat tercapai apabila salah satunya yaitu informasi mengenai bencana yang akurat dari sumber yang terpercaya disampaikan secara cepat dan tepat pada sasaran yang membutuhkan. Sumber yang terpercaya sangat diperlukan untuk menghindari informasi yang menyesatkan masyarakat, penyampaian secara cepat dan tepat sasaran sangat diperlukan agar masyarakat waspada dan ada waktu yang cukup untuk melakukan penyelamatan. Penyampaian informasi secara cepat dapat menggunakan perangkat handphone. Penelitian ini menerapkan algoritma Dijkstra pada handphone untuk mencari lintasan terpendek pada graf berarah [8]

Pembangunan pusat kesehatan kota Balikpapan yakni Rumah Sakit dan Puskesmas mengalami kemajuan yang pesat. Hingga tahun 2010 terdapat 13 Rumah Sakit dan 26 Puskesmas yang tersebar di Kota Balikpapan. Bagi warga kota Balikpapan, khususnya pendatang masih banyak yang belum mengetahui letak-letak bangunan pusat kesehatan tersebut. Hal ini dapat memperlambat dalam penanganan pasien yang sedang kritis. Untuk mengatasi hal tersebut maka dibuatlah sistem penentuan rute terpendek menuju pusat kesehatan dengan menentukan titik awal dan titik tujuan kemudian dibandingkan semua kemungkinan hasil pencarian sehingga mendapatkan rute yang paling terpendek dengan tampilan web geographic information system. Dengan adanya penentuan rute terpendek menuju pusat kesehatan menggunakan metode Dijkstra berbasis webgis diharapkan pengguna dapat dengan mudah mendapatkan informasi mengenai pusat kesehatan yang terdekat untuk memperkecil resiko yang tidak diinginkan [2]

Penerapan algoritma Dijkstra dalam kehidupan sehari-hari semakin besar manfaatnya bagi kehidupan manusia. Salah satu contoh adalah pada saat terjadi gempa bumi pada tanggal 27 Mei 2006 yang melanda DI Yogyakarta dan sebagian Jawa Tengah masih menyisakan duka mendalam bagi masyarakat korban gempa. Ditinjau secara geologis, kepulauan Indonesia berada pada pertemuan dua jalur gempa utama yaitu gempa Alpide Transasiatic sehingga kepulauan Indonesia berada pada daerah yang mempunyai aktifitas gempa bumi cukup tinggi. Sistem Informasi Geografis (SIG) yang dibangun mampu mengambil peta yang tersimpan dalam basis data dan menampilkan peta daerah rawan gempa secara visual dengan format XML SVG pada client serta mampu mencari rute terpendek untuk evakuasi korban gempa dengan algoritma Dijkstra. Sistem Informasi Geografis Pemetaan Daerah Rawan Gempa terdiri dari layer Kabupaten, layer Kecamatan, layer Jalan, Layer Lokasi yang berupa titik lokasi serta rute asal dan rute akhir untuk mendapat rute terpendek yang bisa membantu dalam pengambilan keputusan untuk jalur evakuasi korban gempa [6]

3. METODOLOGI PENELITIAN

Metode yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Lokasi Penelitian
Penelitian dilakukan pada area pelayanan pengambilan sampah yang dilakukan oleh dinas cipta karya di Kota Merauke
2. Sampel Data
 - a. Jumlah Kendaraan atau armada pengangkut sampah
 - b. Kapasitas kendaraan atau armada pengangkut sampah
 - c. Jumlah Konsumen atau tempat pembuangan sementara sampah
 - d. Jumlah permintaan konsumen atau volume tempat pembuangan sementara sampah
 - e. Lokasi konsumen atau tempat pembuangan sementara sampah
 - f. Lokasi tempat pembuangan akhir sampah
 - g. Biaya transportasi yaitu berupa ongkos bahan bakar
3. Prosedur penelitian
 - a. Pengkajian algoritma Dijkstra untuk menentukan rute tercepat pengambilan sampah
 - b. Pengumpulan Data, pada dinas cipta karya kabupaten Merauke
 - c. Pemodelan Algoritma Dijkstra
 - a. Menentukan jarak antar konsumen yang berhubungan secara langsung
 - b. Menggunakan Algoritma Dijkstra mencari jarak terpendek antara tiap-tiap titik terhadap depo
 - c. Mengurutkan titik yang didapat dari proses perhitungan Algoritma Dijkstra
 - d. Pembentukan rute berdasarkan urutan titik
4. Implementasi menggunakan bahasa pemrograman pHp dan MySql sebagai basis data
5. Pengujian digunakan untuk pembuktian bahwa algoritma Dijkstra dapat digunakan sebagai penentu rute tercepat dalam pengambilan sampah

3. HASIL DAN PEMBAHASAN

Analisa Sistem

Untuk menghasilkan rute pengambilan sampah dengan algoritma Dijkstra dilakukan dengan tahap sebagai berikut:

1. Membuat tabel jarak
Tabel ini berisi jarak antara titik yang berhubungan dan menentukan jarak antar titik terhadap depo dan titik lain
 - a. Dibuat tabel antar titik yang saling berhubungan
 - b. Dengan algoritma Dijkstra mencari depo menuju TPA, jarak antar titik yang tidak berhubungan secara langsung dan jarak semua titik terhadap depo.
2. Membentuk rute

Tabel Jarak

Menentukan jarak antar titik yaitu dengan menggunakan panjang masing-masing ruas jalan yang ditentukan dengan melihat peta kota merauke menggunakan MapInfo Professional 7.0. Jarak masing-masing titik yang sudah diketahui kemudian diletakkan dalam tabel jarak.

Tabel Antar Titik Yang Saling Berhubungan

Jalan yang direpresentasikan sebagai sebuah titik (node) ada yang saling berhubungan dan ada yang tidak saling berhubungan. Dalam tahap ini akan dibuat table yang berisi panjang ruas jalan yang diasumsikan jarak antar dua titik (node) atau jarak antara dua jalan yang saling berhubungan.

Contoh perhitungan secara manual dengan menggunakan 10 titik konsumen yaitu berupa 10 TPS yang volume atau permintaannya terakumulasi. Dengan jumlah kendaraan 1 yang mempunyai kapasitas 2 m^3 .

1. Tabel jarak dari jalan yang saling berhubungan dari 10 jalan.

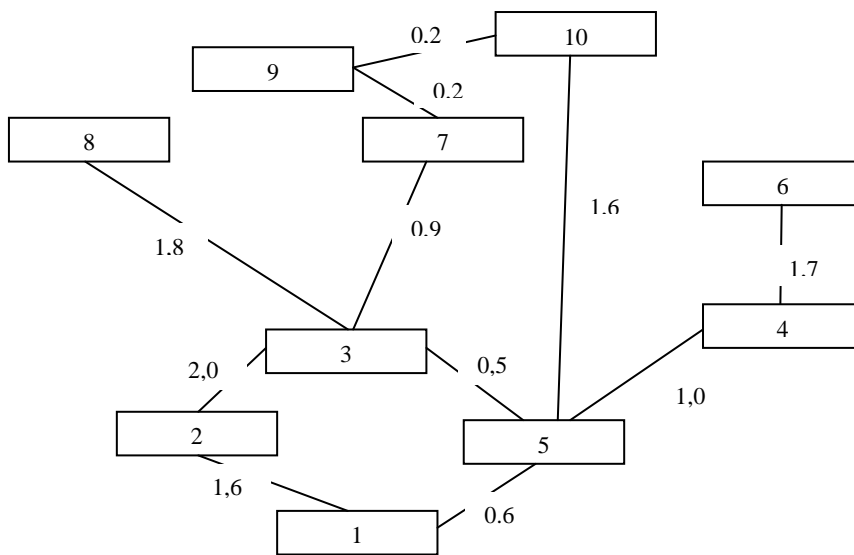
Tabel 1. Titik Saling Berhubungan

	1	2	3	4	5	6	7	8	9	10
1	0	1,60	~	~	0,56	~	~	~	~	~
2	1,60	0	2,06	~	~	~	~	1,81	~	~
3	~	2,06	0	1,01	0,56	~	0,98	1,81	~	1,5
4	~	~	1,01	0	0,56	1,77	~	~	~	1,69
5	0,56	~	0,56	0,56	0	~	~	~	~	1,69
6	~	~	~	1,77	~	0	~	~	~	~
7	~	~	0,98	~	~	~	0	~	0,26	~
8	~	1,81	1,81	~	~	~	~	0	~	~
9	~	~	~	~	~	~	0,26	~	0	0,26
10	~	~	1,5	1,69	1,69	~	~	~	0,26	0

Keterangan

1. Jl. Sesate.
2. Jl. Gak.
3. Jl. Irian Seringgu.
4. Jl. Parako.
5. Jl. Mandala II.
6. Jl. Ermasu.
7. Jl. Kampung Timor.
8. Jl. Ternate.
9. Jl. Timor.
10. Jl. Mandala I.

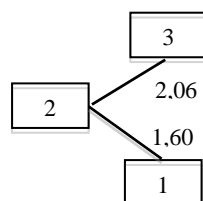
2. Mencari jarak antar titik dengan menggunakan algoritma Dijkstra



Gambar 1. Titik TPS/Konsumen

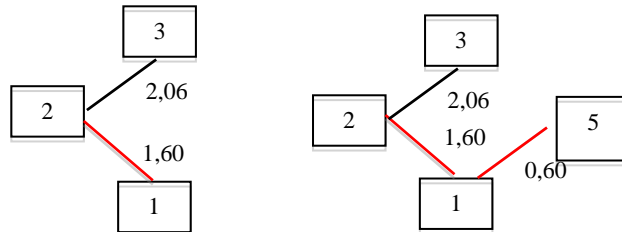
Dari Gambar 1, rancangan penyelesaian dengan algoritma Dijkstra adalah sebagai berikut:

a. Jarak dari Jl. Gak Ke Jl. Madala II:



$$2 \rightarrow 1 = 1,60 \quad 2 \rightarrow 3 = 2,06$$

Jarak yang terkecil adalah 1,60 Maka titik yang dipilih adalah 2 → 1. Kemudian tambahkan dalam daftar nilai yang terhubung dengan titik yang terpilih (titik 1).



Daftar jarak yang harus dipilih bertambah, yaitu dari 1 → 5 = 0,60. Sehingga daftarnya menjadi :

- 2 → 1 = 1,60
- 2 → 3 = 2,06
- 1 → 5 = 0,60

Nilai yang dipilih adalah yang mempunyai nilai terkecil yaitu 1 → 5 = 0,60. Sehingga nilai jarak dari 2 → 5 = 1,60 + 0,60 = 2,20.

	Irian seringgu	sesate
gak	2,06	1,60

	mandala
Gak-irian seringgu	2,06 + 0,56 = 2,62
Gak-sesate	1,60 + 0,60 = 2,20

Jadi jarak terpendek dari jl. Gak ke jl mandala adalah 2,20 melalui jl. Sesate.

- b. Jarak antara jl. Gak ke jl. Parako :
Karena yang terhubung dengan jl. Parako hanya jl. Mandala, maka perhitungan jarak terdekat dapat dimulai dari jalan mandala dengan bobot terkecil. Yaitu 2,20 + 1,02 = 3,22 melalui jl. sesate dan jl. mandala.
- c. Jarak antara Jl. Gak ke Jl. Ermasu
Dihitung dari Jl. Parako. Yaitu 3,22 + 1,77 = 4,95
Dari table tersebut akan dicari jarak antar semua titik yang tidak saling berhubungan dan jarak semua titik terhadap depo dengan menggunakan algoritma dijkstra.

Tabel 2. Jarak

	0	1	2	3	4	5	6	7	8	9	10
0	0	14,48	16,08	14,44	14,90	13,88	16,67	15,44	16,25	15,70	15,57
1	14,48	0	1,60	1,16	1,62	0,56	3,39	2,14	2,97	2,40	2,29
2	16,08	1,60	0	2,06	3,22	2,20	4,99	3,04	1,81	3,30	3,56
3	14,44	1,16	2,06	0	1,01	0,56	3,35	0,98	1,81	1,24	1,5
4	14,90	1,62	3,22	1,01	0	0,56	1,77	2,56	3,39	2,82	1,69
5	13,88	0,56	2,20	0,56	0,56	0	2,79	1,54	2,37	1,80	1,69
6	16,67	3,39	4,99	3,35	1,77	2,79	0	4,33	5,10	4,59	4,48
7	15,44	2,14	3,04	0,98	2,56	1,54	4,33	0	2,79	0,26	0,52
8	16,25	2,97	1,81	1,81	3,39	2,37	5,10	2,79	0	3,05	3,31
9	15,70	2,40	3,30	1,24	2,82	1,80	4,59	0,26	3,05	0	0,26
10	15,57	2,29	3,56	1,5	1,69	1,69	4,48	0,52	3,31	0,26	0

Implementasi

Pada bagian pengaturan jarak ini akan disajikan dalam bentuk matrik. Kolom dan baris yang di isi hanya pada titik yang saling berhubungan secara langsung dan memberikan nilai -1 pada titik yang tidak berhubungan secara langsung.

Tidak semua jalan terhubung secara langsung terhadap jalan lain. Jadi kegunaan matrik ini untuk memberikan nilai yang berupa jarak antara jalan yang terhubung secara langsung dengan jalan lain. Jika jalan tidak terhubung secara langsung maka akan diberikan nilai -1.

Jika matrik sudah terbentuk maka yang akan bekerja kemudian adalah algoritma dijkstra. Algoritma ini akan mencari jalan atau rute terpendek antar depo dan antar semua titik.

	Depo	R.Mandala	G.Aru	G.Nusantara	Biak	Missi	Martadinata	yobar	Onggatmit	TMP	Gal
R.Mandala	1.93	0	0.35	2.42	1.48	2.08	3.64	4.16	2.73	1.56	1.13
G.Aru	2.28	0.35	0	1.56	2.16	2.76	4.32	3.70	2.68	0.87	1.56
G.Nusantara	3.14	2.42	1.56	0	0.6	1.20	2.76	4.68	4.75	3.98	3.64
Biak	3.74	1.48	2.16	0.6	0	0.6	2.16	4.83	3.81	3.04	2.68
Missi	4.34	2.08	2.76	1.20	0.6	0	1.56	6.40	5.90	5.09	4.75
Martadinata	5.90	3.64	4.32	2.76	2.16	1.56	0	7.96	7.46	6.69	6.34
yobar	5.28	4.16	3.70	4.68	4.83	6.40	7.96	0	5.68	4.91	4.56
Onggatmit	4.26	2.73	2.68	4.75	3.81	5.90	7.46	5.68	0	5.56	1.13
TMP	3.49	1.56	0.87	3.98	3.04	5.09	6.69	4.91	5.56	0	2.76
Gak	3.13	1.6	1.55	3.62	2.68	4.77	6.33	4.55	1.13	2.75	0
G.Kangguru	2.77	0.84	1.19	2.17	2.32	3.89	5.45	5.72	3.17	2.39	2.42

Gambar 2. Matriks Jarak Terpendek

```

for ($i = 1; $i <= $this->jumlah_titik; $i++)
    for ($j = 0; $j < $i; $j++)
        if ($this->matriks_jarak[$i][$j] == -1)
            {
                $this->reset_min_jarak();
                $this->reset_bendera();
                $this->temp_titik_asal=$i;
                $this->reset_temp_kamus_hubungan();
                $this->kunjungan($i, $j, 0);
            }
//set jarak terpendek dari koordinat i,j (mirroring)
$this->matriks_jarak_terpendek[$i][$j] = $this->min_jarak;
$this->matriks_jarak_terpendek[$j][$i] = $this->min_jarak;
    
```

Gambar 3. Kode Program Menghitung Jarak Terpendek

7. PENUTUP

Simpulan

Berdasarkan pembahasan yang telah diuraikan pada bab-bab sebelumnya, maka diambil beberapa kesimpulan sebagai berikut:

1. Kombinasi antara algoritma *Dijkstra* dengan metode *saving heuristic* dapat diimplementasikan dengan baik dan dapat menghasilkan rute pengambilan pengambilan sampah di kota Merauke.
2. Penggunaan algoritma *Dijkstra* untuk pemrosesan awal pada penentuan jarak terdekat antar titik konsumen dapat bekerja dengan baik.
3. Metode *saving heuristic* yang dikombinasikan dengan metode eksak yaitu algoritma *dijkstra* ataupun algoritma *brute force* akan menghasilkan waktu proses yang lebih cepat daripada metode eksak.
4. Kombinasi antara metode eksak yaitu algoritma *Dijkstra* dengan metode *heuristic* yaitu *Saving Heuristic* dapat menghasilkan rute yang memenuhi kriteria permasalahan penentuan rute pengambilan sampah, yaitu terdapat depot dimana kendaraan berangkat dan pulang, tiap konsumen tepat dilayani satu kali dalam sebuah rute, kapasitas yang diangkut dalam setiap rute tidak lebih dari kapasitas maksimal kendaraan pengangkut.

Saran

Berdasarkan pada pengujian yang telah dilakukan pada sistem yang dibuat, masih banyak kekurangan dan kelemahan sehingga perlu dikembangkan lagi agar kinerjanya lebih baik, oleh karena itu saran yang diberikan adalah sebagai berikut:

1. Untuk penelitian lebih lanjut diharapkan adanya variasi perpaduan metode sehingga menghasilkan total jarak tempuh yang lebih kecil pada tiap rute yang terbentuk beserta ditambahkan batasan berupa lama waktu pelayanan pada tiap rute.

2. Untuk penelitian lebih lanjut diharapkan adanya perbaikan dalam pembuatan *software* mengenai tampilan dan fasilitas yang lebih lengkap

DAFTAR PUSTAKA

- [1] Ahuja, R.K., T.L. Magnanti dan J.B. Orlin., *Network Flow: Theory Algorithms and Application*, Prentice Hall, New Jersey., 1993.
- [2] Anisiyah, Wiwik., Agus. Fahrul., dan Hamdani, “Penentuan Rute Terpendek Menuju Pusat Kesehatan Menggunakan Metode Dijkstra Berbasis WebGis (Studi Kasus Kota Balikpapan)”, *Jurnal Informatika Mulawarman*, Vol.6 No.3, pp.126-131, September 2011.
- [3] Ballou dan Ronald, H., *Business Logistics Management*, Prentice Hall, Upper Saddle River N.J. 1999.
- [4] Dewi, Luh Joni Erawati., “Pencarian Rute Terpendek Tempat Wisata Di Bali Dengan Menggunakan Algoritma Dijkstra, Prosiding SNATI 2010, pp D-46-D-49, ISSN:1907-5022, Yogyakarta, 19 Juni 2010.
- [5] Fitria, L., Susanty, S. dan Suprayogi, Penentuan Rute Pengambilan dan Pengangkutan Sampah di Bandung, Fakultas Teknologi Industri, Institut Teknologi Bandung, 2009.
- [6] Iskandar, Edi. Dan Hartati, Sri., “Sistem Informasi Geografis Untuk Pemetaan Daerah Rawan Gempa Tektonik Dan Jalur Evakuasi Di Yogyakarta”, *Jurnal Penelitian IPTEK-KOM*, Vol.14 No.1, pp.27-41, Juni 2012.
- [7] Kristian, Y. dan Chaerul, M., “Analisis Awal Implementasi Tempat Pengolahan Sampah Terpadu”, *Skripsi*, Fakultas Teknik Sipil dan Lingkungan, ITB, Bandung, 2010.
- [8] Nadiroh, Siti., Haryanto, dan Munawir, Hafidh., “Implementasi Algoritma Dijkstra Sebagai Solusi Efektif Pembuatan Sistem Bantuan Bencana Real Time”, *Jurnal Ilmiah Teknik Industri*, Vol.12 No.2, pp. 223-234, ISSN: 1424-6869, Desember 2013
- [9] Salaki, Deiby T., “Penentuan Lintasan Terpendek Dari FMIPA Ke Rektorat Dan Fakultas Lain di Unsrat Manado Menggunakan Algoritma Dijkstra”, *Jurnal Ilmiah Sains*, Vol.11 No.1, pp. 73-76, April 2011.