

IMPLEMENTASI ALGORITMA PSO UNTUK PROBABILITAS URUTAN PENGIRIMAN PAKET PENGANTARAN KURIR

Shah Khadafi

Jurusan Sistem Komputer, Institut Teknologi Adhi Tama Surabaya

ABSTRACT

Expanding a shipping company or package delivery services today can not be separated from the role of a courier. Where a courier is main perpetrator of an expedition company whose job to send some packets to the destination addresses in one delivery. Sometimes a courier not yet understand an areas around of Surabaya city which become the delivery of some packet destination address. Therefore, required speed and a good understanding of an area of Surabaya city for a courier when sending some of its packets delivery. Besides that, sometimes a courier are confused to determine the sequence of delivery of packets in one day delivery. These problems can result package is not delivered which the schedules of package delivery predetermined. Issues about courier is related with the probability of the sequence of package delivery address. One algorithm that produces the probability is the PSO algorithm that a continuation of Particle Swarm Optimization. With PSO algorithm can generate multiple solutions to determine the sequence of the fastest delivery of the addresses of several package. The solutions that generated by PSO algorithm can be used as a reference by a courier for delivering of several packages to some destination addresses.

Keywords: Distance, Optimize, Path, PSO, Shipping

ABSTRAK

Berkembangnya sebuah perusahaan ekspedisi atau jasa pengiriman paket saat ini tidak lepas dari peran serta seorang kurir. Dimana kurir merupakan pelaku utama dari sebuah perusahaan ekspedisi yang bertugas untuk mengirimkan beberapa paket ke beberapa alamat tujuan dalam satu pengiriman. Terkadang seorang kurir belum memahami area-area sekitar kota Surabaya yang menjadi daerah pengiriman dari beberapa alamat tujuan paket pengirimannya. Maka dituntut kecepatan dan pemahaman yang baik sebuah area kota Surabaya bagi seorang kurir ketika mengirimkan beberapa paket-paketnya. Selain itu kadang kala kurir merasa kebingungan untuk menentukan urutan-urutan pengiriman dari paket-paket dalam satu hari pengiriman. Permasalahan tersebut dapat mengakibatkan paket tidak terkirim sesuai dengan jadwal yang telah ditentukan sebelumnya. Permasalahan yang menyangkut kurir terkait dengan probabilitas urutan-urutan alamat pengantaran paket. Salah satu algoritma yang menghasilkan probabilitas adalah algoritma PSO yang kepanjangan dari Particle Swarm Optimization. Dengan algoritma PSO ini dapat menghasilkan solusi untuk menentukan urutan-urutan tercepat pengiriman alamat-alamat dari beberapa paket. Solusi-solusi yang dihasilkan oleh algoritma PSO dapat dijadikan acuan oleh kurir untuk proses pengiriman beberapa paket ke beberapa alamat tujuan.

Kata kunci: Jarak, Jalur, Optimasi, Pengiriman, PSO

PENDAHULUAN

Salah satu faktor yang terpenting dari perusahaan ekspedisi ini yaitu, kurir. Seorang kurir seorang kurir dituntut agar mengirimkan paket dalam waktu yang sangat singkat, khususnya ketika kurir tersebut mencari alamat tujuan dari paket tersebut. Seorang kurir juga dituntut memahami sebuah area sebuah kota tertentu khususnya di kota Surabaya. Algoritma *Particle Swarm Optimization* (PSO) diperkenalkan oleh J.Kennedy dan R.C.eberhart. Algoritma PSO terinspirasi perilaku hewan-hewan yang berdifat sosial organisme misalkan, dari sekawanan serangga, seperti semut, rayap, ataupun lebah ataupun sekelompok burung. Perilaku sosial bisa juga dari perilaku ataupun tindakan oleh sebuah individu dalam suatu kelompok. Misalkan perilaku burung dalam kawanan burung ketika sedang terbang. Sifat utama implementasi dari PSO adalah pengaturan kecepatan partikel secara probabilistik dan heuristic. Probabilitas adalah suatu nilai yang dihasilkan dari beberapa peluang atau kemungkinan yang terjadi secara random. Nilai probabilitas sering dinyatakan dalam bilangan desimal (seperti 0,50, 0,20 atau 0,89). Nilai dari probabilitas berkisar antara 0 sampai dengan 1.

TINJAUAN PUSTAKA

Travelling Salesman Problem (TSP)

TSP permasalahan optimasi yang menarik perhatian para peneliti, dikarenakan kasus TSP mudah didefinisikan namun sulit diselesaikan. Mengambil contoh permasalahan seorang sales yang harus melewati beberapa kota tujuan yang setiap kota tujuan hanyalah dikunjungi satu kali dan selanjutnya kembali ke kota asal dan total jarak yang ditempuh haruslah semimumimum mungkin. Permasalahan TSP dalam kasus ini diselesaikan menggunakan algoritma PSO.

Particle Swarm Optimization (PSO)

Setiap partikel diasumsikan berperilaku terdistribusi yang sifatnya bisa dipengaruhi dari perilaku kawanan makhluk tersebut. *Swarm* pada PSO mengacu pada populasi yang mempunyai sekelompok kawanan tertentu yang menuju (mencari) ke sumber makanan. Setiap partikel disini diasumsikan mempunyai dua karakteristik, yang pertama *position* : setiap partikel disini menyampaikan informasi atau posisi bagusya kepada partikel yang lain lalu mengubah posisi mereka terhadap waktu. Dan yang kedua *velocity*: setiap partikel menyesuaikan posisi dan kecepatan berdasarkan informasi yang diterima mengenai posisi partikel sebelumnya.

Google Maps API

Google Maps adalah peta digital yang cukup populer, yang dikembangkan oleh *developer* Google. Google Maps API adalah sebuah *library* yang dikembangkan dengan menggunakan kode bahasa pemrograman *JavaScript*. System yang dikembangkan ini memanfaatkan *library* Google Maps API yang digunakan untuk, diantaranya :

1. Memperoleh alamat valid yang dapat dibaca oleh server Google Maps yang digunakan untuk memperoleh nilai *latitude* maupun *longitude* dengan Geocoding API.
2. Memperoleh nilai jarak dari perhitungan jarak antara alamat asal dan alamat tujuan dengan Distance Matrix API.

Geocoding API

Google Maps *Geocoding* API adalah layanan yang menyediakan *geocoding* dan mengembalikan alamat *geocoding*[1]. Proses didalam *geocoding* API ini proses mengkonversi alamat menjadi koordinat geografis. Untuk proses konversi didalam API ini, yaitu disisipkan kedalam bahasa pemrograman *JavaScript* kemudian memanggil fungsi dari API, dengan syntax, `geocoder.geocode({'address':address},function(result,status))`. Sebagai contoh alamat yang hendak diketahui alamat valid dengan request ke server Google Maps, yaitu :

1. Inputan alamat : "Jl.Raya Arjuna no.75 Surabaya".
2. Alamat kembalian dari Google Maps : "Jl. Raya Arjuna No.75, Kec. Sawahan, Kota SBY, Jawa Timur 60251, Indonesia".



Gambar 1. Nilai Latitude dan Longitude Sebuah Alamat

Geometry API

Google Maps *JavaScript Geometry* API *library* menyediakan fungsi utilitas untuk perhitungan data geometris pada permukaan bumi[2]. *Object geometry library* memiliki properti *location* yang

digunakan menghasilkan nilai *latitude* dan *longitude* sebuah alamat. Sehingga nilai *geometry* yang didapat dipisahkan *latitude* dan *longitude*. Contoh penggunaan *geometry* library :

- a. `results[0].geometry.location.lat()`
- b. `results[0].geometry.location.lng()`

Contoh untuk alamat “Jl. Raya Arjuna No.75, Kec. Sawahan, Kota SBY, Jawa Timur 60251, Indonesia”, nilai-nilai koordinatnya yaitu:

- a. *latitude* -7.262679
- b. *longitude* 112.72700499999996

Distance Matrix API

Google Maps *Distance Matrix* API adalah layanan yang memberikan jarak perjalanan dan waktu perjalanan dari matriks asal dan tujuan, yang mana berdasarkan pada rute yang direkomendasikan antara titik awal dan akhir[3]. Beberapa layanan-layanan yang disediakan oleh *Distance Matrix* API diantaranya, yaitu :

a. Distance matrix request

Layanan ini memiliki berbagai macam mode perjalanan yang disediakan dalam layanan Google's *Distance Matrix*. Parameter rekues yang dilakukan oleh user dengan menggunakan layanan *distance matrix* ini dapat dituliskan seperti berikut:
[https://maps.googleapis.com/maps/api/distancematrix/json?origins=""&destinations=""&language=id-ID&key=AIzaSyDSgn_6_xgUOhhXIflYmtl9DtLJoUC-vpw](https://maps.googleapis.com/maps/api/distancematrix/json?origins=).

b. Distance matrix response

Didalam layanan *distance matrix response* terdapat beberapa elemen yang menyesuaikan dengan keinginan user melalui request parameter yang disisipkan kedalam bahasa pemrograman *JavaScript*. Elemen-elemen pada *distance matrix response* diantaranya, yaitu:

1. *origin_addresses* : berisi sebuah array alamat asal yang dikembalikan oleh API berasal dari rekues user. Array ini diformat oleh *object geocoder* yang diterjemahkan sesuai dengan parameter *language* melalui *request parameter*.
2. *destination_addresses* : berisi sebuah array alamat tujuan yang dikembalikan oleh API berasal dari rekues user. Array ini diformat oleh *object geocoder* yang diterjemahkan sesuai dengan parameter *language* melalui *request parameter*. Misal:
 - `$data['origin_addresses']`=Jl. Raya Arjuna No.75, Kec. Sawahan, Kota SBY, Jawa Timur 60251, Indonesia.
 - `$data['destination_addresses']`=Ngagel Tirta IV No.80, Wonokromo, Kota SBY, Jawa Timur 60245, Indonesia.
 - Maka parameter rekues untuk mendapatkan jarak dan waktu tempuh antara alamat asal dan alamat tujuan, yaitu :
https://maps.googleapis.com/maps/api/distancematrix/json?origins=Jl.%20Raya%20Arjuna%20No.65,%20Tegalsari&destinations=Jl.%20Ngagel%20Tirta%20IV%20No.80,%20Kec.%20Sawahan&language=id-ID&key=AIzaSyDSgn_6_xgUOhhXIflYmtl9DtLJoUC-vpw
 - Nilai hasil parameter rekues tersebut, yaitu :
 - o `$data['rows'][0]['elements'][0]['distance']['value']` =6657
 - o `$data['rows'][0]['elements'][0]['duration']['text']` = 18 menit

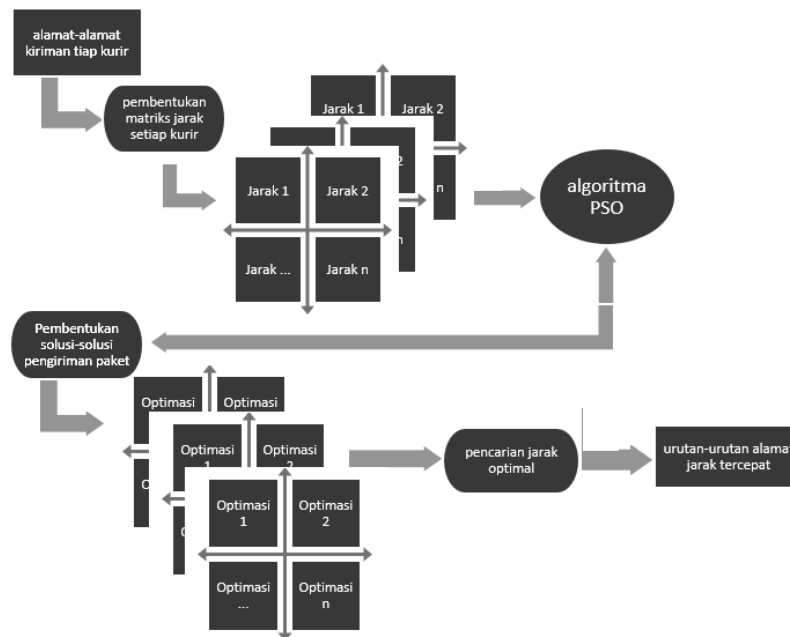
METODE

Perancangan Sistem

Rancangan sistem ini nampak pada penelitian ini, mengenai tahapan desain sistem akan dibagi menjadi 3 tahapan, diantaranya yaitu :

1. Pra-proses.
2. Pembentukan matriks jarak.
3. Pembentukan solusi-solusi urutan pengiriman paket.

Untuk proses perancangan system secara keseluruhan nampak seperti pada gambar 2.



Gambar 2. Blok Diagram Rancangan Sistem

HASIL DAN PEMBAHASAN

Pra Proses

Tahap pra proses ini bertujuan mendapatkan data-data yang valid tentang beberapa alamat-alamat tujuan pengantaran paket oleh kurir. Data-data yang didapatkan nantinya yaitu, berupa nilai jarak, waktu, nilai *latitude* dan *longitude* dari alamat tujuan pengantaran paket. Proses-proses yang terlibat didalam tahap pra proses, yaitu : Input data dan penyimpanan data-data ke dalam *database* : Hasil inputan dari user dan alamat-alamat valid yang dapat dibaca oleh Google maps disimpan ke dalam database. Database yang digunakan untuk system ini menggunakan database MySQL.

Tabel 1. Contoh Alamat Tujuan Kurir

No	Alamat Tujuan	Jarak	Lat	Long
0	Jl. Raya Arjuna No.75, Kec. Sawahan, Kota SBY, Jawa Timur 60251, Indonesia	0	-7.262	112.727
1	Jl. Raya Rungkut Lor No.10, Tenggilis Mejoyo, Kota SBY, Jawa Timur 60299, Indonesia	15611	-7.309	112.769
2	Jl. Siwalankerto Timur I No.10, Wonocolo, Kota SBY, Jawa Timur 60236, Indonesia	11631	-7.338	112.738
3	Jl. Medayu Utara XXX No.19, Rungkut, Kota SBY, Jawa Timur 60295, Indonesia	16175	-7.321	112.803
4	Jl. Pandugo Tim. XIII, Rungkut, Kota SBY, Jawa Timur 60297, Indonesia	15611	-7.318	112.789
5	Jl. Rungkut Permai IV, Gn. Anyar, Kota SBY, Jawa Timur 60293, Indonesia	13458	-7.331	112.772

Pembentukan Matriks Jarak

Matriks jarak dibuat dengan ukuran $n \times n$, dimana n adalah menunjukan baris dan kolom. Data-data didalam matriks jarak tersebut adalah nilai jarak antar alamat asal dan alamat tujuan yang dilalui oleh seorang kurir. Pada sistem ini, untuk mendapatkan nilai jarak didalam matriks jarak memanfaatkan layanan *Distance Matrix* API untuk me-request jarak antara alamat awal (alamat asal) dengan alamat tujuan.

Tabel 2. Matriks Jarak

-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	11001	10088	13775	12150	11465	4960	12509	9050	6745	8582	11462	12287	13909	11815	3869	8481
1	11001	0	7425	5357	3732	3554	7104	4598	1588	2856	2691	3550	4855	5490	4970	7073	1450
2	10088	7425	0	9748	8752	5374	9748	6418	7729	6753	4947	6385	6319	12045	4781	10423	6312
3	13775	5357	9748	0	2197	4494	13432	5303	7026	8808	8643	4639	4507	3945	7581	12846	7402
4	12150	3732	8752	2197	0	3978	11400	4787	4993	6776	6611	4123	3991	4155	5549	10813	5370
5	11465	3554	5374	4494	3978	0	10967	1743	5210	6477	6312	1305	1150	3002	2788	10694	5071
6	4960	7104	9748	13432	11400	10967	0	10329	6870	4565	6402	9282	10107	11728	9634	746	6301
7	12509	4598	6418	5303	4787	1743	10329	0	5863	7130	6982	1958	614	2230	2991	11348	5724
8	9050	1588	7729	7026	4993	5210	6870	5863	0	3707	3542	3848	4673	6295	5268	6353	2301
9	6745	2856	6753	8808	6776	6477	4565	7130	3707	0	1806	5701	6526	8148	6200	5201	2720
10	8582	2691	4947	8643	6611	6312	6402	6982	3542	1806	0	4991	5932	7635	4394	7732	3402
11	11462	3550	6385	4639	4123	1305	9282	1958	3848	5701	4991	0	1434	3332	3451	10085	4462
12	12287	4855	6319	4507	3991	1150	10107	614	4673	6526	5932	1434	0	1852	3842	10734	5110
13	13909	5490	12045	3945	4155	3002	11728	2230	6295	8148	7635	3332	1852	0	4921	12169	6726
14	11815	4970	4781	7581	5549	2788	9634	2991	5268	6200	4394	3451	3842	4921	0	10512	5704
15	3869	7073	10423	12846	10813	10694	746	11348	6353	5201	7732	10085	10734	12169	10512	0	5923
16	8481	1450	6312	7402	5370	5071	6301	5724	2301	2720	3402	4462	5110	6726	5704	5923	0

Keterangan :

- Khusus indeks baris dan indeks kolom no 0 : merupakan tempat atau alamat awal pemberangkatan kurir.
- Baris i , mengindikasikan indeks alamat asal.
- Kolom j , mengindikasikan indeks alamat tujuan.
- Nilai-nilai pada matriks jarak adalah nilai jarak yang didapatkan dari alamat asal ke alamat tujuan. Nilai jarak didapatkan dengan merekues ke Google maps.

Pembentukan Solusi-solusi Urutan Pengiriman Paket

Pembentukan solusi-solusi pengiriman disini memanfaatkan algoritma PSO yang nilai-nilainya menghasilkan beberapa probabilitas. Probabilitas yang dihasilkan nantinya mempunyai nilai optimasi berdasarkan dari partikel terbaik ($pbest$) yang dihasilkan algoritma PSO. Sedangkan masing-masing partikel terbaik dituliskan dengan, $pbest_i = \{pbest_1, pbest_2, pbest_3, \dots, pbest_n\}$. Matriks optimasi dibuat dengan ukuran $n \times n$ (sama dengan dimensi matriks jarak). Nilai yang terdapat didalam matriks jarak dihitung per-baris (0 – 16 baris) dengan algoritma PSO. Nilai probabilitas algoritma PSO digunakan membentuk urutan-urutan pengiriman paket untuk kurir.

Sebelum dilakukan penghitungan algoritma PSO, yaitu menentukan parameter-parameter awal, yaitu :

- $c1 = c2 = 1$
- $r1 = 0,8$; $r2 = 0,5$
- $W = 1$
- Iterasi = 100

Dari matriks PSO, kemudian dilakukan proses penentuan urutan-urutan pengiriman paket kurir. Untuk menentukan urutan alamat-alamat pengiriman dengan memperhatikan sebuah nilai optimal dari $P_{best,i}$ pada masing-masing baris atau masing-masing swarm ($swarm-0, swarm-2, swarm-3, s/d swarm-16$).

Tabel 3. Matriks Algoritma PSO

-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	10000	-5,258	-4,840	-6,596	-5,818	-5,490	-2,383	-5,968	-4,331	-3,227	-4,097	-5,488	-5,876	-6,663	-5,665	-1,886	-4,046
1	-46,595	10000	-44,893	-43,908	-43,134	-43,049	-44,740	-43,546	-28,377	-42,715	-42,638	-43,047	-43,668	-43,971	-43,723	-44,725	-950
2	-4,840	-3,567	10000	-4,680	-4,182	-2,590	-4,680	-3,063	-3,719	-3,231	-2,376	-3,047	-3,014	-5,775	-2,293	-4,974	-3,010
3	-6,596	-2,581	-4,680	10000	-1,050	-2,150	-6,425	-2,554	-3,367	-4,210	-4,127	-2,222	-2,156	-1,924	-3,645	-6,132	-3,555
4	-5,818	-1,817	-4,182	-1,050	10000	-1,940	-5,457	-2,296	-2,399	-3,242	-3,160	-1,981	-1,947	-1,989	-2,677	-5,164	-2,588
5	-52,032	-48,265	-49,132	-48,713	-48,466	10000	-51,796	-46,700	-49,053	-49,657	-49,578	-978	-45,284	-48,001	-47,900	-51,666	-48,986
6	-28,996	-30,016	-31,276	-33,031	-32,062	-31,856	10000	-31,552	-29,905	-28,807	-29,681	-31,054	-31,447	-32,218	-31,221	-559	-29,633
7	-47,733	-43,966	-44,834	-44,303	-44,056	-40,396	-46,697	10000	-44,569	-45,172	-45,102	-41,218	-39,455	-42,838	-43,201	-47,182	-44,504
8	-32,077	-595,25	-31,447	-31,112	-30,143	-30,248	-31,037	-30,559	10000	-29,533	-29,453	-29,599	-29,990	-30,763	-30,274	-30,791	-28,860
9	-34,788	-32,938	-34,792	-35,771	-34,803	-34,658	-33,748	-34,969	-33,341	10000	-677	-34,291	-34,681	-35,457	-34,530	-34,051	-32,871
10	-35,661	-32,857	-33,935	-35,691	-34,723	-34,585	-34,623	-34,904	-33,265	-677	10000	-33,955	-34,401	-35,216	-33,666	-35,258	-33,197
11	-52,031	-48,263	-49,612	-48,782	-48,535	-978	-50,993	-46,725	-48,404	-49,288	-48,949	10000	-46,663	-48,159	-48,215	-51,374	-48,697
12	-46,895	-43,354	-44,052	-43,190	-42,944	-862	-45,857	-40,011	-43,268	-44,151	-43,869	-41,144	10000	-41,193	-42,873	-46,156	-43,478
13	-39,002	-34,993	-38,115	-34,261	-34,356	-33,808	-37,965	-33,440	-35,375	-36,262	-36,014	-33,970	-694	10000	-34,722	-38,175	-35,585
14	-5,665	-2,388	-2,293	-3,645	-2,677	-1,345	-4,623	-1,447	-2,537	-2,968	-2,100	-1,677	-1,872	-2,363	10000	-5,013	-2,755
15	-28,476	-30,001	-31,596	-32,751	-31,783	-31,726	-559	-32,038	-29,658	-29,110	-30,317	-31,436	-31,746	-32,429	-31,640	10000	-29,454
16	-29,393	-543	-28,360	-28,880	-27,911	-27,769	-28,354	-28,080	-26,449	-26,649	-26,975	-27,479	-27,788	-28,557	-28,071	-28,175	10000

Maka solusi urutan pengiriman paket berdasarkan indek untuk kurir yaitu :
0-15-9-16-8-10-1-11-12-7-5-14-2-4-3-13-6. Dengan jarak tempuh 61856 Meter.

Probabilitas Urutan-urutan Pengiriman Paket

Dengan mengganti nilai r (r1 dan r2) parameter algoritma PSO hasil akhirnya ditampilkan dalam probabilitas-probabilitas urutan pengiriman kurir.

Tabel 4. Probabilitas Urutan Pengiriman Paket Kurir

Probabilitas ke-	Urutan Pengiriman Ke-																	Total Jarak
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	0	15	9	16	8	10	1	11	12	7	5	14	2	4	3	13	6	61856
2	0	16	1	2	12	5	3	13	6	9	10	7	8	11	14	15	4	87568
3	0	13	3	7	12	4	14	5	11	1	2	8	10	16	9	6	15	71083

KESIMPULAN

- Pada algoritma PSO tidak adanya pengaturan parameter untuk mengatur kecepatan konvergen partikel (CE coefficient) menjadikan nilai optimization terjebak dalam local optima.
- Pengaturan parameter yang berbeda-beda untuk r1, r2 maka setiap setiap probabilitas menghasilkan urutan pengiriman yang berbeda-beda.
- Distance matrix API dapat digunakan hanya untuk request jarak dari alamat asal ke alamat tujuan. Request jarak oleh distance matrix API ini terbatas hanya sampai 2500 request per harinya.

DAFTAR PUSTAKA

- [1] Google. Google Maps APIs. Distance Matrix API, Getting Started. <https://developers.google.com/maps/documentation/distance-matrix/start>. 18 Oktober 2016.
- [2] Kennedy, J., Eberhart, R. 1995. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*. IV (pp. 1942-1948).
- [3] Xuesong Yan , Can Zhang , Wenjing Luo , Wei Li , Wei Chen and Hanmin Liu. November 2012. Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm. *IJCSI International Journal of Computer Science Issues*, Vol. 9, Issue 6, No 2.