

# APLIKASI FIREFLY ALGORITHM UNTUK PEMECAHAN MASALAH TRAVELLING SALESMAN PROBLEM: GEDUNG KAMPUS INSTITUT TEKNOLOGI SUMATERA

*Laporan Akhir Swarm Intelligence*

## **Kelompok 7**

Ketua Kelompok : 120450009 - Alfian Dwi Kurnia

Anggota 1 : 120450015 - Alfa Khoirin

Anggota 2 : 120450081 - Dimas Wahyu Saputro

Anggota 3 : 120450069 - Rezki P. Manullang

## INTRODUCTION

*Travelling Salesman Problem* (TSP) merupakan masalah matematika untuk menemukan jarak terpendek untuk mengunjungi seluruh kota, tepat hanya sekali. Penyelesaian untuk masalah TSP ini mengharuskan perhitungan terhadap semua kemungkinan rute yang dapat diperoleh, kemudian memilih rute yang terpendek. Terdapat banyak algoritma yang bisa digunakan, yang mana masing-masing algoritma memiliki kekurangan dan kelebihan masing-masing. Salah satu algoritma yang bisa digunakan yaitu *Firefly Algorithm* (FA).

FA merupakan salah satu contoh komputasi yang terinspirasi dari alam untuk mencari solusi optimisasi baru. Kebanyakan dari komputasi yang terinspirasi dari alam akan menggunakan kecerdasan koloni. Contoh penerapan TSP menggunakan FA, adalah kasus mencari rute terpendek untuk mengunjungi seluruh gedung yang ada di ITERA. Dengan menggunakan TSP melalui pendekatan *Firefly Algorithm*, dapat dicari solusi paling optimal mengenai hal ini.

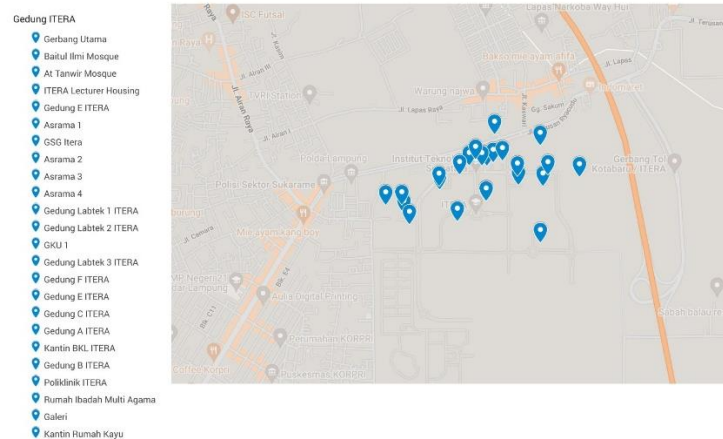
## PROBLEM STATEMENT

*Travelling Salesman Problem* merupakan salah satu masalah yang dapat diselesaikan menggunakan algoritma kunang-kunang. Tujuan utama dari TSP adalah mencari jarak terpendek untuk mengunjungi seluruh titik, tepat hanya sekali, dan kembali ke titik awal. Salah satu contoh kasus yang dapat diselesaikan dengan TSP adalah kasus Gedung-gedung yang ada di Institut Teknologi Sumatera (ITERA).

ITERA merupakan perguruan tinggi baru yang memiliki luas hampir 275 Ha, dan memiliki gedung, kebun raya, embung, taman, parkir, dan tempat lainnya. Menggunakan TSP melalui pendekatan *Firefly Algorithm*, kasus ini dapat diselesaikan dengan mudah dan akan didapatkan rute paling pendek sehingga seseorang dapat mengunjungi seluruh titik yang ada. Namun, penulis membatasi ruang lingkup percobaan kali ini pada seluruh gedung yang ada di ITERA. Bayangkan jika seseorang hanya mempunyai waktu satu hari, dan ingin mengelilingi seluruh gedung yang ada. Pasti sangat melelahkan jika mengunjungi seluruh gedung tanpa arah dan tanpa pola.

## DATA DESCRIPTION

Institut Teknologi Sumatera (ITERA) merupakan kampus baru yang berada di Lampung Selatan. Meskipun baru, ITERA memiliki gedung yang terbilang tidak sedikit. Jika dilakukan pengukuran secara manual akan menghabiskan waktu yang cukup banyak. Oleh karena itu, penulis menggunakan *Google Maps*. Penulis menandai gedung yang berada di ITERA.



Gambar 1.1 Gedung yang ada di ITERA

Dilakukan penandaan sebanyak 24 titik, yang tersebar dari gerbang, masjid, gedung perkuliahan, laboratorium, dan gedung lainnya. Setelah dilakukan penandaan, akan didapatkan data Latitude dan Longitude masing-masing titik. Bentuk awal yang didapatkan bukan berupa *.csv*, melainkan bentuk *.kml*. Oleh karena itu, data harus diolah supaya bentuknya menjadi *.csv* dan bisa digunakan menjadi data pada *Research Based Learning* ini. Data yang penulis dapatkan saat ini, penulis namakan **Data Awal**.

city	lat	lon
Asrama 1	105.3174256	-5.3585498
Asrama 2	105.3193365	-5.3584711
Asrama 3	105.3190301	-5.3591681
Asrama 4	105.3174999	-5.3591123
At Tanwir Mosque	105.3188553	-5.356638
Baitul Ilmi Mosque	105.3125303	-5.3591874
Galeri	105.3155174	-5.3579165
Gedung A ITERA	105.3144117	-5.357889
Gedung B ITERA	105.3152146	-5.357901
Gedung C ITERA	105.3138196	-5.3584598
Gedung D ITERA	105.3133344	-5.3588016
Gedung E ITERA	105.3154738	-5.3600983
Gedung F ITERA	105.3136715	-5.3613616
Gedung Labtek 1 ITERA	105.3102016	-5.3603317
Gedung Labtek 2 ITERA	105.3106715	-5.3615497
Gedung Labtek 3 ITERA	105.3091909	-5.3603426
Gerbang Utama	105.3148152	-5.3575059
GUKU 1	105.3103228	-5.3608767
GSG Itera	105.3188646	-5.3626901
Wisma	105.3160008	-5.3559504
Kantin BKL ITERA	105.3164947	-5.3575952
Kantin Rumah Kayu	105.3125606	-5.3593812
Poliklinik ITERA	105.3159198	-5.3577029
Rumah Ibadah Multi Agama	105.321311	-5.358592

Tabel 1.1 Data awal lokasi gedung ITERA

Data awal terdiri dari 25 baris, dan 3 kolom. Kolom pertama *city* merupakan kolom untuk nama-nama gedung yang ada di ITERA. Selanjutnya, kolom *lat* dan *lon* berisi data *latitude* dan *longitude* masing-masing titik.

city	lat	lon
Asrama 1	105.3174256	-5.3585498
Asrama 2	105.3193365	-5.3584711
Asrama 3	105.3190301	-5.3591681
Asrama 4	105.3174999	-5.3591123
At Tanwir Mosque	105.3188553	-5.356638
Baitul Ilmi Mosque	105.3125303	-5.3591874
Galeri	105.3155174	-5.3579165
Gedung A ITERA	105.3144117	-5.357889
Gedung B ITERA	105.3152146	-5.357901
Gedung C ITERA	105.3138196	-5.3584598
Gedung D ITERA	105.3133344	-5.3588016
Gedung E ITERA	105.3154738	-5.3600983
Gedung F ITERA	105.3136715	-5.3613616
Gedung Labtek 1 ITERA	105.3102016	-5.3603317
Gedung Labtek 2 ITERA	105.3106715	-5.3615497
Gedung Labtek 3 ITERA	105.3091909	-5.3603426
Gerbang Utama	105.3148152	-5.3575059
GKU 1	105.3103228	-5.3608767
GSG Itera	105.3188646	-5.3626901
Wisma	105.3160008	-5.3559504
Kantin BKL ITERA	105.3164947	-5.3575952
Kantin Rumah Kayu	105.3125606	-5.3593812
Poliklinik ITERA	105.3159198	-5.3577029
Rumah Ibadah Multi Agama	105.321311	-5.358592

Tabel 1.1 Data awal lokasi gedung ITERA

Data awal hanya berisi *latitude* dan *longitude* masing-masing titik. Oleh karena itu, menggunakan matriks ketetanggaan dan *Haversine Vectorize Function*, penulis mendapatkan jarak antar titik. Penulis namakan data saat ini, yaitu **Data Jarak Titik**. Data jarak antar lokasi gedung terdiri dari 25 baris, dan 25 kolom. *Cell* yang berwarna hitam berarti jaraknya 0 KM. Hal ini disebabkan karena *longitude* dan *latitude*-nya sama. Jarak yang didapatkan disini dalam skala Kilo Meter (KM), yang berarti jika didapatkan angka 0.1 KM, artinya sama dengan 100 Meter.

Nama Gedung	Asrama 1	Asrama 2	Asrama 3	Asrama 4	At Tanwir Mosque	Baitul Ilmi Mosque	Galeri	Gedung A ITERA	Gedung B ITERA	Gedung C ITERA	Gedung D ITERA	Gedung E ITERA	Gedung F ITERA	Gedung Labtek 1 ITERA	Gedung Labtek 2 ITERA	Gedung Labtek 3 ITERA	Gerbang Utama	GKU 1	GSG Itera	Wisma	Kantin BKL ITERA	Kantin Rumah Kayu	Poliklinik ITERA	Rumah Ibadah Multi Agama
Asrama 1		0.212562	0.179391	0.018479	0.168656	0.544826	0.2131	0.335797	0.246667	0.401103	0.455124	0.221815	0.425661	0.805228	0.756409	0.917457	0.2919698	0.7929949	0.201048	0.1759	0.1073	0.54168	0.169328	0.43217
Asrama 2	0.212562		0.039762	0.205152	0.075938	0.757345	0.4251	0.548052	0.458784	0.613644	0.667683	0.4323	0.635814	1.017543	0.968041	1.129832	0.5037025	1.0050801	0.134634	0.3784	0.3171	0.75416	0.380709	0.21065
Asrama 3	0.179391	0.039762		0.170212	0.07685	0.722972	0.3924	0.515077	0.426027	0.570937	0.633623	0.39651	0.59591	0.982587	0.932355	1.094956	0.4713596	0.9698114	0.105122	0.35	0.2858	0.71963	0.348627	0.25427
Asrama 4	0.018479	0.205152	0.170212		0.167377	0.552772	0.2233	0.346375	0.256673	0.408908	0.463418	0.227217	0.430929	0.812579	0.76289	0.934915	0.3023255	0.7998885	0.184647	0.1909	0.1204	0.54943	0.180567	0.42418
At Tanwir Mosque	0.168656	0.075938	0.07685	0.167377		0.707595	0.3732	0.495626	0.406651	0.562672	0.61373	0.389623	0.593061	0.968647	0.921649	1.080464	0.4501026	0.9572029	0.177849	0.3181	0.2641	0.70478	0.328011	0.27012
Baitul Ilmi Mosque	0.544826	0.757345	0.722972	0.552772	0.707595		0.3343	0.212715	0.300956	0.144993	0.090155	0.328497	0.142097	0.261192	0.218086	0.372987	0.2589048	0.250503	0.712039	0.3976	0.4434	0.00662	0.379527	0.97683
Galeri	0.213063	0.42511	0.392444	0.223295	0.37317	0.334346		0.122989	0.033683	0.189519	0.244203	0.064284	0.228099	0.595517	0.549473	0.707295	0.0790316	0.584301	0.397853	0.0789	0.1091	0.33169	0.045197	0.64473
Gedung A ITERA	0.335797	0.548052	0.515077	0.345375	0.495626	0.212715	0.123		0.089307	0.06796	0.12279	0.134793	0.131093	0.473754	0.429694	0.585164	0.0462709	0.4631974	0.514992	0.1857	0.2319	0.231051	0.167835	0.76769
Gedung B ITERA	0.246667	0.458784	0.426027	0.256673	0.406651	0.300956	0.0337	0.089307		0.156032	0.210801	0.070701	0.199488	0.562148	0.516567	0.673843	0.0459166	0.5510904	0.429683	0.1046	0.1427	0.29839	0.078655	0.67841
Gedung C ITERA	0.401103	0.613644	0.579937	0.409808	0.562672	0.144993	0.1895	0.06796	0.156032		0.054895	0.190189	0.086821	0.406168	0.361734	0.517811	0.1142311	0.3953742	0.574754	0.2536	0.2986	0.14263	0.234661	0.83328
Gedung D ITERA	0.455124	0.667683	0.633623	0.463418	0.61737	0.090155	0.2442	0.12279	0.210801	0.054895		0.240995	0.084031	0.351347	0.306996	0.463098	0.1690501	0.3404798	0.625643	0.3082	0.3533	0.08774	0.289379	0.88726
Gedung E ITERA	0.221815	0.4323	0.39651	0.227217	0.389623	0.328497	0.0643	0.134793	0.070701	0.190189	0.240995		0.203876	0.586466	0.535858	0.698883	0.1056744	0.5734009	0.38477	0.1352	0.1353	0.32472	0.086104	0.65078
Gedung F ITERA	0.425661	0.635814	0.59951	0.430929	0.593061	0.142097	0.2289	0.131093	0.199488	0.086821	0.084031	0.203876		0.38714	0.333735	0.499274	0.170333	0.3727476	0.578945	0.304	0.3329	0.13657	0.272199	0.85363
Gedung Labtek 1 ITERA	0.805228	1.017543	0.982587	0.812579	0.968647	0.261192	0.5955	0.473754	0.562148	0.406168	0.351347	0.586466	0.38714		0.063337	0.11242	0.5198396	0.0209272	0.966072	0.6578	0.7046	0.26387	0.640705	1.23675
Gedung Labtek 2 ITERA	0.756409	0.968041	0.932355	0.76289	0.921649	0.218086	0.5495	0.429694	0.516567	0.361734	0.306996	0.535858	0.333735	0.063337		0.168459	0.4759636	0.0435321	0.911933	0.6152	0.658	0.21957	0.594604	1.18662
Gedung Labtek 3 ITERA	0.917457	1.129832	1.094956	0.924915	1.080464	0.372987	0.7072	0.585164	0.673843	0.517811	0.463098	0.698883	0.499274	0.11242	0.168459		0.6311144	0.1258744	1.078212	0.7684	0.8164	0.37587	0.75246	1.3491
Gerbang Utama	0.29197	0.503702	0.47136	0.302326	0.450103	0.258905	0.079	0.046271	0.045917	0.114231	0.16903	0.105674	0.170332	0.515984	0.475964	0.631114		0.5094044	0.475472	0.1396	0.1868	0.25676	0.123001	0.72232
GKU 1	0.792995	1.00508	0.969811	0.799889	0.957203	0.250503	0.5843	0.463197	0.55109	0.395374	0.34048	0.573401	0.372748	0.020927	0.043532	0.126874	0.5094044		0.951595	0.6479	0.6932	0.25278	0.629496	1.22406
GSG Itera	0.201048	0.134634	0.105122	0.184647	0.177849	0.712039	0.3979	0.514992	0.429683	0.574754	0.625643	0.38477	0.578945	0.966072	0.911933	1.078212	0.4754735	0.9515954		0.3751	0.3031	0.7079	0.358835	0.29757
Wisma	0.175924	0.37835	0.349962	0.190879	0.318148	0.397564	0.0789	0.185705	0.106554	0.253568	0.308185	0.13523	0.303971	0.657758	0.615174	0.768373	0.139568	0.6479299	0.37508	0.0732	0.3957	0.052272	0.59573	
Kantin BKL ITERA	0.107276	0.317139	0.285774	0.120367	0.264071	0.443433	0.1091	0.231852	0.142668	0.298633	0.353302	0.135289	0.332949	0.704581	0.658048	0.816399	0.1868289	0.6932342	0.303148	0.0732	0.44072	0.064024	0.53652	
Kantin Rumah Kayu	0.541684	0.754156	0.719629	0.549454	0.704782	0.006616	0.3317	0.210512	0.298389	0.14263	0.087737	0.324719	0.136575	0.263872	0.219566	0.375873	0.2567583	0.2527563	0.7079	0.3957	0.4407	0.376882	0.97358	
Poliklinik ITERA	0.169328	0.380709	0.348627	0.180567	0.328011	0.379527	0.0452	0.167835	0.078655	0.234661	0.289379	0.086104	0.272199	0.640705	0.594604	0.75246	0.1230007	0.6294956	0.358835	0.0523	0.064	0.37688	0.60023	
Rumah Ibadah Multi Agama	0.432174	0.219652	0.254268	0.424184	0.279118	0.976832	0.6447	0.767686	0.67405	0.833276	0.887257	0.650778	0.853626	1.236753	1.186616	1.349097	0.7232312	1.2240575	0.297574	0.5957	0.5365	0.97358	0.600231	

Tabel 1.2 Data jarak antar lokasi gedung ITERA

## METHODS

### 1. Deskripsi umum Tentang Metode Swarm Intelligence (SI) yang digunakan

Metode yang digunakan oleh kelompok kami adalah metode Algoritma *Firefly* dengan kasus *Travelling Salesman Problem* (TSP). Algoritma *Firefly* adalah algoritma metaheuristik yang memperoleh inspirasi dari cara berkedipnya kunang – kunang. TSP adalah permasalahan umum dalam optimasi kombinatorial dimana seorang salesman harus mengunjungi sejumlah N kota, disyaratkan setiap kota hanya dikunjungi sekali. Salesman ini harus memilih rute sehingga jarak total yang di tempuh minimum.

Pada *Firefly Algorithm*, Seekor kunang-kunang akan mencoba meningkatkan solusi yang diwakilinya dengan bergerak lebih dekat ke kunang-kunang lain dengan lebih baik nilai kebugaran, di ruang pencarian. Proses ini diulang sampai kondisi berhenti tercapai. Kinerja dari pendekatan yang diusulkan adalah benchmark dengan studi kasus.

Algoritma *Firefly* diperkenalkan oleh X.-S. Yang pada tahun 2007 yang pada dasarnya didasarkan pada perkawinan perilaku kunang-kunang. Kunang-kunang dengan intensitas cahaya flash yang lebih rendah akan menarik kunang-kunang lainnya dengan intensitas senter yang lebih besar. Seperti SI lainnya, X.-S. Yang memodelkan senter sebagai kebugaran dari algoritma dan posisi kunang-kunang dalam ruang pencarian sebagai representasi dari kandidat solusi.

Dalam pemodelan rute TSP, posisi kunang-kunang di FA mewakili kandidat solusi masalah. Setiap dimensi posisi kunang-kunang mewakili kota TSP. Sebuah kota akan menjadi kota pertama yang dilalui oleh salesman jika memiliki suara terbesar dalam hal nilai numerik, kota kedua adalah kota dengan terbesar kedua pemungutan suara dan sebagainya.

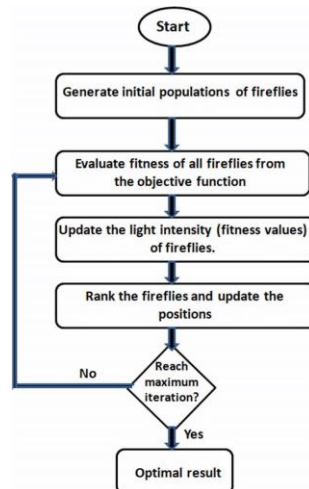
## 2. Algoritma/Pseudocode dari Metode SI yang digunakan

```
Initialize all the parameters ( $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $n$ );
Initialize randomly a population of fireflies;
Evaluate the fitness of the initial population at  $x_i$  by  $f(x_i)$  for  $i = 1, \dots, n$ ;
while ( $t < \text{MaxGeneration}$ ) do
    for All fireflies ( $i = 1 : n$ ) do
        for All other fireflies ( $j = 1 : n$ ) (inner loop) do
            if Firefly  $j$  is better/brighter than  $i$  then
                Move firefly  $i$  towards  $j$  according to Eq. (1);
            end
        end
        Evaluate the new solution and accept the new solution if better;
    end
    Rank and update the best solution found so far;
    Update iteration counter  $t \leftarrow t + 1$ ;
    Reduce  $\alpha$  (randomness strength) by a factor;
end
```

## 3. Penjelasan Mengenai Modifikasi dari metode yang digunakan

Algoritma Firefly dibuat untuk menyelesaikan masalah continuous optimization. Untuk menyelesaikan masalah discrete optimization, diperlukan teknik untuk mendiskritisasi.

## 4. Flowchart dari metode yang digunakan dalam RBL



Gambar 1.2 Diagram alir Travelling Salesman Problem Menggunakan Firefly Algorithm

## RESULTS AND DISCUSSIONS

### 1. Solusi terbaik, parameter terbaik, dan iterasi berhenti.

#### a. Solusi terbaik

Travelling Salesman Problem (TSP) merupakan salah satu permasalahan optimasi yang terjadi di kehidupan sehari-hari. Permasalahan pada kasus TSP adalah bagaimana membangun rute terpendek yang akan dilalui salesman. Setelah memvariasikan parameter, penulis mendapatkan solusi terbaik yang bisa didapatkan menggunakan algoritma Firefly. Solusi terbaik terjadi ketika parameter  $\gamma = 20$ . Visualisasi gambar dapat dilihat pada gambar 3.1.

Urut	Nama Titik	Kode Titik
1	Galeri	6
2	Gedung F ITERA	12
3	GKU 1	17
4	Gedung Labtek 1 ITERA	13
5	Poliklinik ITERA	22
6	At Tanwir Mosque	4
7	Gedung B ITERA	8
8	Rumah Ibadah Multi Agama	23
9	Gedung E ITERA	11
10	Gedung Labtek 2 ITERA	14
11	Gedung Labtek 3 ITERA	15
12	Baitul Ilmi Mosque	5
13	Gedung D ITERA	10
14	Gedung C ITERA	9
15	Gedung A ITERA	7
16	GSG Itera	18
17	Asrama 4	3
18	Asrama 1	0
19	Kantin Rumah Kayu	21
20	Wisma	19
21	Asrama 3	2
22	Asrama 2	1
23	Gerbang Utama	16
24	Kantin BKL ITERA	20

Tabel 3.1 Solusi Terbaik

Penulis mencoba menggabungkan seluruh parameter terbaik yang didapatkan dari analisis yang sudah dilakukan. Parameternya yang digunakan seperti tabel di bawah ini.

	n_individu	n_kota	a	b	max_generasi	alpha	beta0	gamma
1	100	24	1	10	100	30	20	20

Tabel 3.2 Parameter Gabungan

Setelah dijalankan, hasilnya tidak sesuai dengan perkiraan penulis. Jarak yang didapatkan adalah 7.842114. Nilai parameter gamma ketika 20 masih lebih besar dari pada jika menggunakan parameter gabungan.

b. Parameter terbaik

Pada percobaan kali ini, terdapat tiga parameter yang penulis variasikan, yaitu  $\beta_0$ ,  $\alpha$ , dan  $\gamma$  (Intensitas cahaya saat  $r=0$ , step size, koefisien absorpsi). Pergerakan kunang-kunang sebagian besar digambarkan oleh formula:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha (rand - \frac{1}{2}),$$

atau,

$$\beta = \beta_0 \cdot e^{-\gamma r}$$

$$x_i = x_i \cdot (1 - \beta) + x_j \cdot \beta + \alpha (rand - \frac{1}{2}),$$

Kedua formula di atas sama dan equivalent. Jika kita perhatikan dari formula, semakin besar  $\beta_0$  dan  $\alpha$ , intensitas cahaya saat  $r = 0$ , dan step size, akan berbanding lurus dengan  $x_i$ , pergerakan kunang-kunang. Namun hal ini tidak sama dengan parameter  $\gamma$ , atau koefisien absorpsi. Koefisien absorpsi ( $\gamma$ ) menentukan seberapa besar daya tarik berkurang oleh jarak. Penulis memvariasikan parameter yang ada sebagai berikut:

	n_individu	n_kota	a	b	max_ge nerasi	alpha	beta0	gamma
1	50	24	1	10	100	0.01	1	1
2	50	24	1	10	100	0.1	1	1
3	50	24	1	10	100	1	1	1
4	50	24	1	10	100	10	1	1
5	50	24	1	10	100	20	1	1
6	50	24	1	10	100	30	1	1
7	50	24	1	10	100	50	1	1
8	50	24	1	10	100	100	1	1
9	50	24	1	10	100	1	0.01	1
10	50	24	1	10	100	1	0.1	1
11	50	24	1	10	100	1	1	1
12	50	24	1	10	100	1	2	1
13	50	24	1	10	100	1	10	1
14	50	24	1	10	100	1	20	1
15	50	24	1	10	100	1	30	1
16	50	24	1	10	100	1	50	1
17	50	24	1	10	100	1	1	0.01
18	50	24	1	10	100	1	1	0.1
19	50	24	1	10	100	1	1	1
20	50	24	1	10	100	1	1	2
21	50	24	1	10	100	1	1	10
22	50	24	1	10	100	1	1	20
23	50	24	1	10	100	1	1	30
24	50	24	1	10	100	1	1	50

Tabel 3.3 Parameter Metode

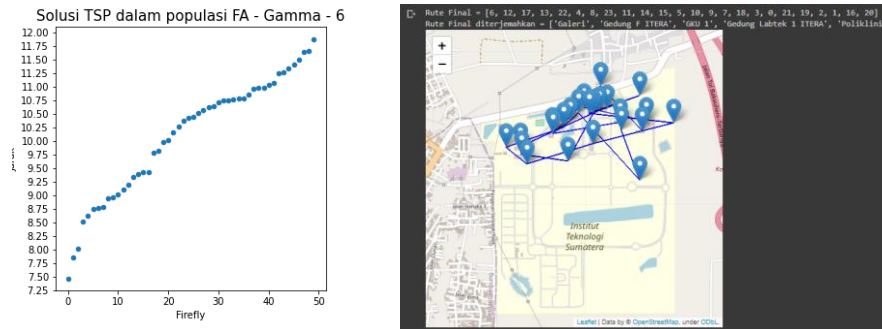


Setelah parameter pada tabel 3.3 digunakan, akan menghasilkan solusi yang bervariasi. Hasil setiap parameter dapat dilihat pada analisis nomor 2. Hasil terbaik setiap parameter sebagai berikut:

Nama Parameter	Nilai Parameter	Jarak (KM)
Alpha ( $\alpha$ )	30	7.644359388
Beta0 ( $\beta_0$ )	20	7.585667705
Gamma ( $\gamma$ )	20	7.462324779

Tabel 3.4 Hasil Terbaik dari Parameter Metode

Dari percobaan, didapatkan nilai parameter terbaik dari setiap parameter yang bisa dilihat pada tabel 3.4. Jarak terpendek didapatkan ketika nilai Gamma ( $\gamma$ ) adalah 20, dengan jaraknya 7.462324779 KM. Hal yang menarik adalah, pada hasil terbaik parameter Alpha ( $\alpha$ ) dan Beta0 ( $\beta_0$ ), titik mulai adalah pada titik 0, yaitu Asrama 1. Sedangkan pada hasil terbaik parameter gamma ( $\gamma$ ) dimulai pada titik 6, yaitu Galeri, seperti yang terlihat pada gambar 3.1.



Gambar 3.1 Visualisasi Solusi

### c. Iterasi berhenti.

Pada program *Travelling Salesman Problem* (TSP) menggunakan *Firefly Algorithm*, penulis menggunakan pengulangan while. Ketika iterasi kurang dari maksimal generasi, atau kriteria berhenti belum terpenuhi, maka pengulangan akan terjadi.

```

37 def FA(params, df):
38
39     generasi = 0
40     new_kunangs_w_f = inisialisasi(params, df)
41     temp = np.zeros_like(new_kunangs_w_f[:, :-1])
42
43     while generasi < params['max_generasi']:
44
45         for i in range(int(params['n_individu'])):
46             for j in range(int(params['n_individu'])):
47                 if (new_kunangs_w_f[i, -1] > new_kunangs_w_f[j, -1]) and (i != j):
48                     temp[i, :] = movement(new_kunangs_w_f[:, :-1], i, j, params)
49
50         new_kunangs_w_f = sort_individu(calculate_fitness(temp, df))
51         temp = new_kunangs_w_f[:, :-1]
52
53         generasi += 1
54
55     return solusi(new_kunangs_w_f)

```

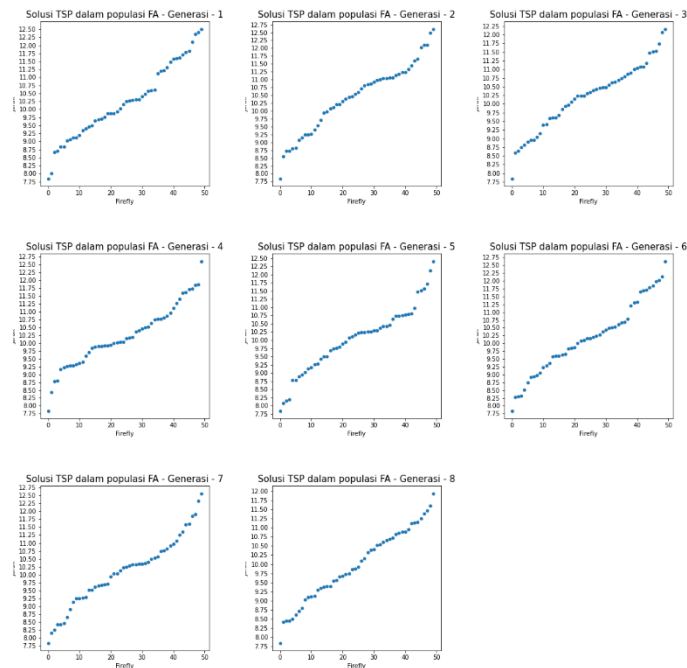
Gambar 3.2 Potongan Kode Program

Oleh karena hal di atas, penulis mencoba memvariasikan maksimal generasi supaya dapat dianalisis seperti tabel di bawah ini:

	n_individu	n_kota	a	b	max_generasi	alpha	beta0	gamma
1	50	24	1	10	1	1	1	1
2	50	24	1	10	2	1	1	1
3	50	24	1	10	3	1	1	1
4	50	24	1	10	5	1	1	1
5	50	24	1	10	10	1	1	1
6	50	24	1	10	15	1	1	1
7	50	24	1	10	20	1	1	1
8	50	24	1	10	50	1	1	1

Tabel 3.5 Parameter Variasi Maksimal Generasi

Dari percobaan memvariasikan parameter maksimal generasi, penulis mencoba memvisualisasikan, dan didapatkan kesimpulan bahwa generasi tidak terlalu mempengaruhi jarak TSP. Terlihat meskipun kami memvariasikan maksimal generasi, jarak terendah selalu pada jarak 7.842114 KM.

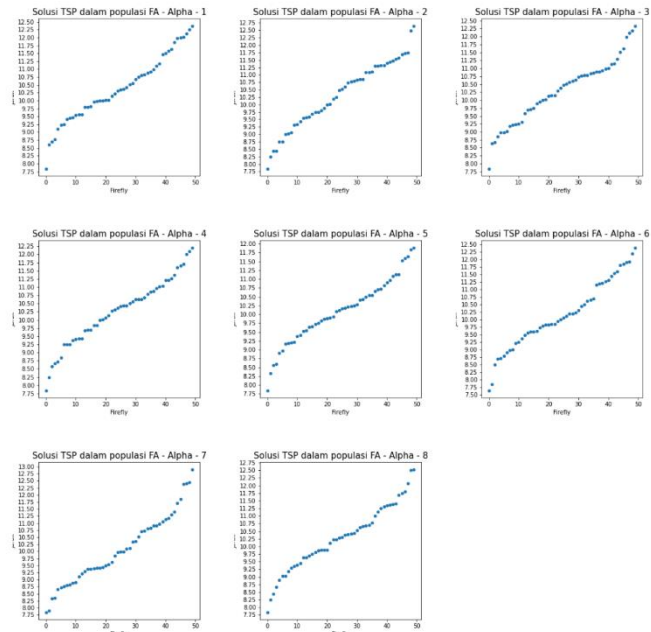


Gambar 3.3 Visualisasi Parameter Maksimal Generasi

## 2. Visualisasi hubungan antara parameter dan solusi terbaik.

### a. Parameter

#### i. Alpha (randomization rate)

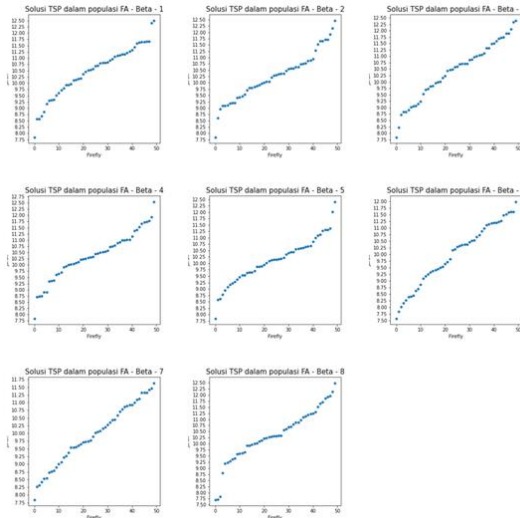


Gambar 3.4 Visualisasi Parameter Alpha

Pada visualisasi di atas kami mendapatkan solusi terbaik pada TSP kunang-kunang dengan memvariasikan parameter alpha 0.01,0.1,1,10,20,30,50 dan 100. Pada variasi dengan solusi terbaik pada alpha ke-6 dengan nilai alpha sebesar 30 dengan jarak 7.644359 KM.

Dari hasil analisa, didapatkan suatu kesimpulan bahwa semakin banyak jumlah firefly yang digunakan, maka nilai deviasi yang didapat akan jauh lebih baik. Siste firefly algorithm membandingkan nilai tiap individu firefly yang ada dalam satu iterasi. Apabila jumlah firefly yang digunakan semakin banyak, maka sarana pembanding nilai objektif akan semakin bervariasi.

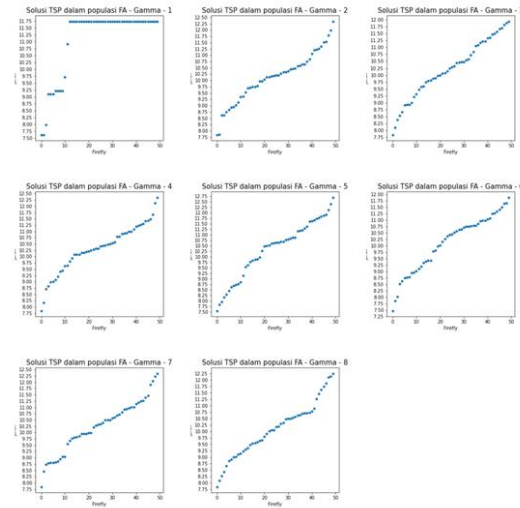
ii. Beta0 (attractive firefly when n)



Gambar 3.5 Visualisasi Parameter Beta

Jika dilihat pada visualisasi di atas didapatkan solusi terbaik pada TSP kunang-kunang dengan merubah parameter Beta0 0.01, 0.1, 1, 2, 10, 20, 30, dan 50 di dapatkan variasi dengan solusi terbaik terdapat pada Beta0 ke-6 dengan nilai Beta0 sebesar 20 dengan jarak 7.58566 KM.

iii. Gamma (coefficient of light)

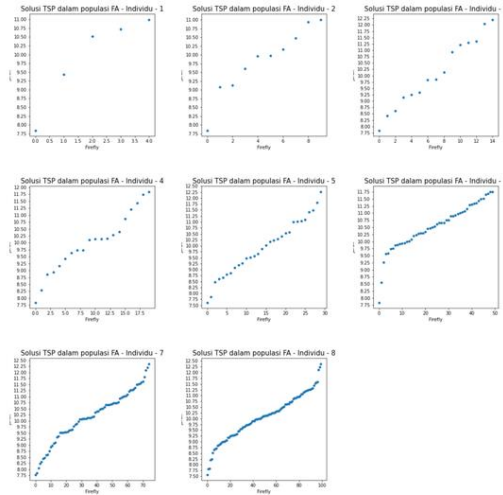


Gambar 3.6 Visualisasi Parameter Gamma

Jika dilihat pada visualisasi diatas didapatkan solusi terbaik pada TSP kunang-kunang dengan merubah parameter Gamma 0.01, 0.1, 1, 2, 10, 20, 30, dan 50 di dapatkan variasi dengan solusi terbaik terdapat pada Gamma ke-6 dengan nilai Gamma sebesar 20 dengan jarak 7.462324779 KM. semakin besar nilai gamma

yang digunakan, maka nilai deviasi yang dihasilkan semakin baik. Hal ini didasarkan pada arti nilai gamma itu sendiri. Semakin besar nilai gamma, maka pergerakan firefly akan semakin berkurang yang menyebabkan hanya ada perbandingan kecerahan saja pada tiap firefly.

#### b. Jumlah Individu

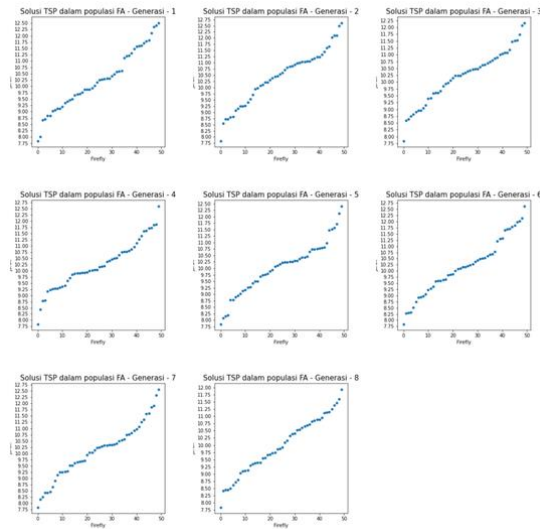


Gambar 3.7 Visualisasi Parameter Individu

Jika dilihat pada visualisasi diatas didapatkan solusi terbaik pada TSP kunang-kunang dengan merubah parameter Jumlah Individu 5, 10, 15, 20, 30, 50, 75 dan 100 di dapatkan variasi dengan solusi terbaik terdapat pada Jumlah Individu ke-8 dengan nilai individu sebesar 100 dengan jarak terpendek 7.568711258 KM.

Dari hasil analisis, didapatkan suatu kesimpulan bahwa semakin banyak jumlah firefly yang digunakan, maka nilai deviasi yang didapat akan jauh lebih baik. Siste firefly algorithm membandingkan nilai tiap individu firefly yang ada dalam satu iterasi. Apabila jumlah firefly yang digunakan semakin banyak, maka sarana pembanding nilai objektif akan semakin bervariasi.

c. Maksimal generasi

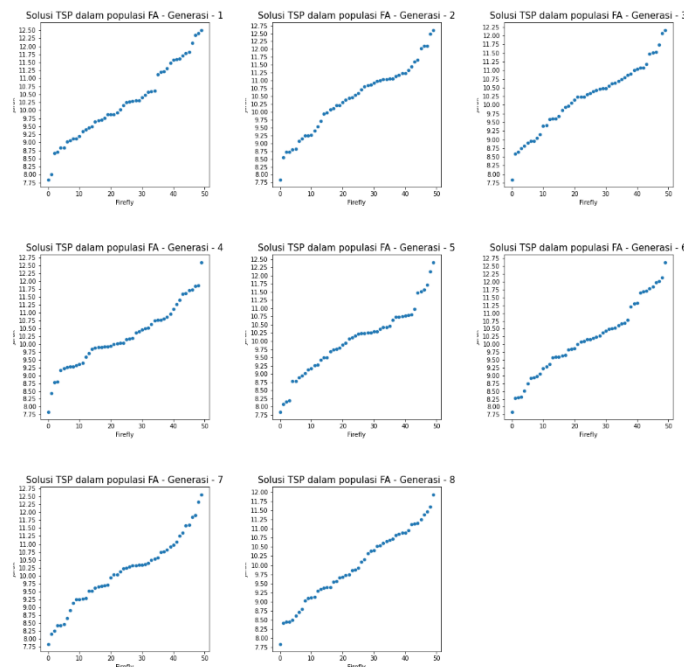


Gambar 3.8 Visualisasi Parameter Maksimal Generasi

Jika dilihat pada visualisasi di atas didapatkan solusi terbaik pada TSP kunang-kunang dengan merubah parameter Jumlah maksimal generasi 1, 2, 3, 5, 10, 15, 20 dan 50. generasi tidak terlalu mempengaruhi jarak TSP. Terlihat meskipun kami memvariasikan maksimal generasi, jarak terendah selalu pada jarak 7.842114 KM.

### 3. Visualisasi solusi terhadap iterasi

Pada pengujian jumlah iterasi, parameter jumlah iterasi dibuat bervariasi nilainya di angka 1 iterasi, 2 iterasi, 3 iterasi, 5 iterasi, 10 iterasi, 15 iterasi, 20 iterasi dan 50 iterasi. Pada parameter firefly algorithm yang lain, yaitu jumlah populasi firefly, alpha, beta0 dan nilai gamma, parameter tersebut dibuat tetap nilainya. Dari percobaan memvariasikan parameter maksimal generasi, penulis mencoba memvisualisasikan, dan didapatkan kesimpulan bahwa generasi tidak terlalu mempengaruhi jarak TSP. Terlihat meskipun kami memvariasikan maksimal generasi, jarak terendah selalu pada jarak 7.842114 KM.



Gambar 3.9 Visualisasi Parameter Iterasi

Dari hasil analisis, didapatkan suatu kesimpulan bahwa semakin banyak iterasi yang digunakan, maka nilai yang didapatkan akan jauh lebih baik. Hal ini didasarkan pada karakteristik firefly algorithm yang membandingkan nilai objektif yang dihasilkan dari setiap iterasi. Semakin banyak iterasi yang digunakan, maka nilai objektif yang dihasilkan akan semakin banyak sesuai banyaknya iterasi. Banyaknya iterasi akan membuat sarana pembandingan nilai objektif yang jauh lebih variatif. Namun karena firefly algorithm merupakan suatu proses komputasi, maka seiring bertambahnya iterasi akan menimbulkan proses komputasi data yang lebih lama.

#### 4. Perbandingan Firefly Algorithm dan Genetic Algorithm

Travelling Salesman Problem merupakan masalah yang dapat diselesaikan dengan berbagai pendekatan algoritma, seperti Firefly Algorithm dan Genetic Algorithm. Pada kasus penulis, gedung kampus Institut Teknologi Sumatera, jika diselesaikan menggunakan Firefly Algorithm, akan didapatkan solusi optimal ketika parameter nilai Gamma sebesar 20 dengan jarak 7.462324779 KM. Penulis tertarik untuk mencoba menyelesaikan kasus menggunakan Genetic Algorithm sehingga bisa dibandingkan dengan hasil FA.

Langkah-langkah untuk menyelesaikan kasus TSP menggunakan Genetic Algorithm, sebagai berikut:

1. Tentukan ruang solusi.
2. Hitung seberapa bagus solusinya, disebut juga fungsi fitness.
3. Secara acak menghasilkan satu set solusi yang mungkin (disebut sebagai populasi awal).
4. Hitung fitness untuk masing-masing solusi ini.
5. Pilih solusi yang “baik” berdasarkan fungsi kebugaran Anda. Dari titik ini, ada banyak cara untuk maju, kita akan membahas tentang mutasi dan cross over, bidang optimasi memiliki banyak cara lain, saya mendorong Anda untuk melakukan penelitian dan melihat luasnya bidang ini.
6. Pilih beberapa solusi dari populasi di atas untuk iterasi berikutnya dari populasi.
7. Hasilkan angka acak dan berdasarkan ambang batas, apakah kami memutasikan solusi atau melakukan crossover (lebih lanjut tentang ini nanti di artikel).
8. Ulangi langkah 4 sampai 7 sebanyak yang Anda bisa berdasarkan waktu yang Anda miliki untuk menemukan solusi terbaik atau daya komputasi.
9. Sepanjang proses, kami melacak solusi terbaik dan mengembalikan solusi terbaik di akhir.



Dari langkah-langkah di atas, penulis mencoba menjalankan program TSP – GA, dengan parameter number\_of\_cities=24, initial\_pop\_size=10000, nelite=10, percentage\_to\_crossover=20, dan percentage\_to\_mutate=20. Didapatkan hasil seperti pada gambar 3.9. Dengan populasi awal 10000, dan jumlah iterasi 100, didapatkan jarak terpendek yaitu 5.99778 KM.

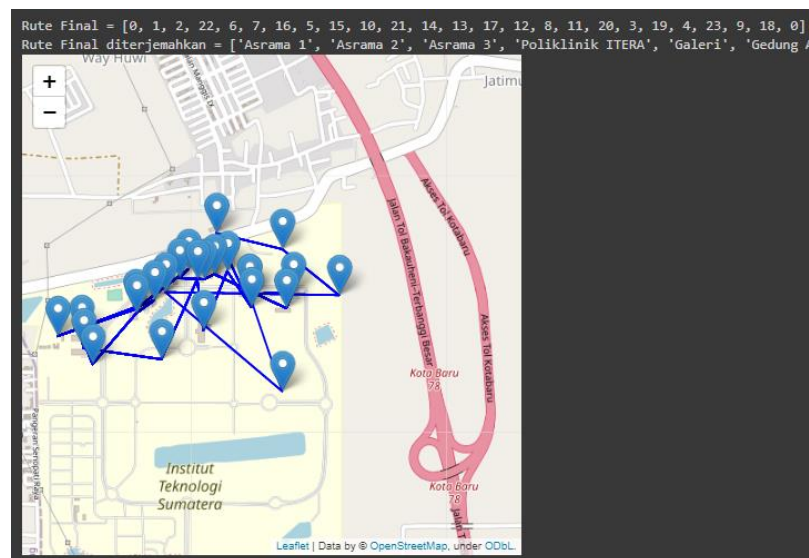
```

Crossover: 100% | 3/3 [00:00<00:00, 2008.77it/s]
Crossover: 100% | 3/3 [00:00<00:00, 1631.60it/s]
Crossover: 100% | 3/3 [00:00<00:00, 1664.85it/s]
-----
Best solution for initial population size of 10000 and number
of runs 100 is
Route      Cost
1 [1, 2, 22, 6, 7, 16, 5, 15, 10, 21, 14, 13, 17... 5.59778
Generated 200(0.0%) of the 25852016738884976640000 solutions
Total distance travelled to cover the final route of
Asrama 1 => Asrama 2 => Asrama 3 => Poliklinik ITERA => Galeri => Gedung A ITERA
(Generated from initial population size of 10000)
-----

```

Gambar 3.9 Hasil Running Program

Berbeda dengan TSP FA yang memiliki solusi optimal ketika parameter nilai Gamma sebesar 20 dengan jarak 7.462324779 KM dan dimulai dari titik 6, yaitu Galeri. TSP GA dimulai dari titik 0, yaitu Asrama 1.



Gambar 3.10 Hasil Visualisasi TSP GA

Jika dibandingkan antara TSP FA dan TSP GA, didapatkan jarak terpendek 5.99778 KM, yaitu ketika menggunakan TSP GA dengan parameter di atas. Hasil yang sangat menarik untuk dipelajari lebih lanjut.

## CONCLUSION

Penerapan *Firefly Algorithm* dalam *Travelling Salesman Problem* berhasil diterapkan pada kasus Gedung Kampus Institut Teknologi Sumatera. Tujuannya untuk mencari jarak terpendek bagi mahasiswa atau masyarakat untuk mengunjungi semua gedung sekaligus. Pada percobaan, didapatkan solusi optimal ketika parameter nilai gamma sebesar 20 dengan jarak 7.462324779 KM. Dari hasil analisis parameter jumlah iterasi, jumlah populasi, maupun penentuan nilai gamma, didapatkan suatu kecenderungan Kecepatan pemrosesan data pada firefly algorithm berbanding terbalik dengan meningkatnya nilai parameter firefly. Semakin kecil nilai parameter yang dipakai, maka kecepatan pemrosesan data akan jauh lebih cepat. Namun apabila digunakan nilai parameter yang jauh lebih besar, maka kecepatan pemrosesan data akan jauh lebih lambat, namun dapat menghasilkan suatu penyelesaian yang lebih baik.

## REFERENCES

- Faisal Mohd Hussein, S., Faiz Zainal Abidin, A., Asraf Bin Mansor, M., & Nor Shah, B. (2013). A Preliminary Study on Firefly Algorithm Approach for Travelling Salesman Problem Piezoelectric micro generation system View project My Research View project. <https://www.researchgate.net/publication/244988875>
- Gilang, K., & Jati, S. (2011). Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem.
- Jati, G. K., Manurung, R., & Suyanto. (2013). Discrete Firefly Algorithm for Traveling Salesman Problem: A New Movement Scheme. In *Swarm Intelligence and Bio-Inspired Computation* (pp. 295–312). Elsevier Inc. <https://doi.org/10.1016/B978-0-12-405163-8.00013-2>
- Yang, X.-S. (Ed.). (2018). *Nature-Inspired Algorithms and Applied Optimization* (Vol. 744). Springer International Publishing. <https://doi.org/10.1007/978-3-319-67669-2>

## APPENDIX

1. <https://linktr.ee/swarm7SD>
2. [https://www.google.com/maps/@-5.3607019,105.3137854,1585m/data=!3m1!1e3!4m2!6m1!1s1SWPxHqh9\\_CO5JAw-Qa4AA4NJMdb90Zvg](https://www.google.com/maps/@-5.3607019,105.3137854,1585m/data=!3m1!1e3!4m2!6m1!1s1SWPxHqh9_CO5JAw-Qa4AA4NJMdb90Zvg)