



ITERA

APLIKASI FIREFLY ALGORITHM UNTUK
PEMECAHAN MASALAH TRAVELING SALESMAN
PROBLEM :

GEDUNG KAMPUS INSTITUT TEKNOLOGI
SUMATERA

Kelompok 7

Alfian Dwi Kurnia

Alfa Khoirin

Dimas Wahyu Saputro

Rezki P.Manullang

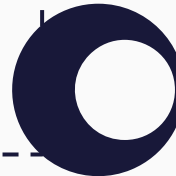
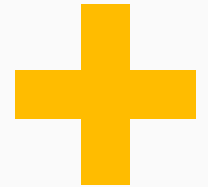


Table of contents



1

Latar Belakang

2

Masalah

3

Deskripsi Data

4

Metode

5

Pemodelan
Individu

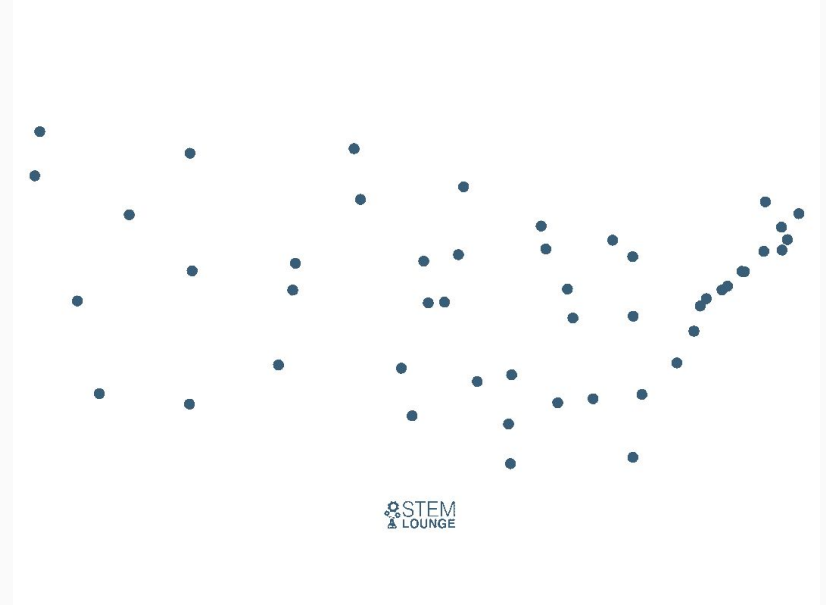
6

Hasil dan Analisis

7

Kesimpulan



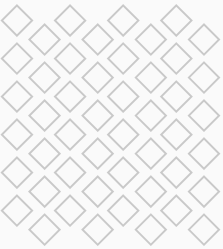


STEM
& LOUNGE

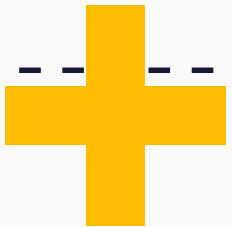





Latar Belakang

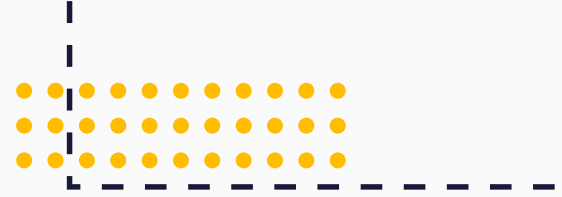


Travelling Salesman Problem (TSP) merupakan masalah matematika untuk menemukan jarak terpendek untuk mengunjungi seluruh kota, tepat hanya sekali. Salah satu algoritma yang bisa digunakan yaitu *Firefly Algorithm* (FA). FA merupakan salah satu contoh komputasi yang terinspirasi dari alam untuk mencari solusi optimisasi baru. 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.





Masalah

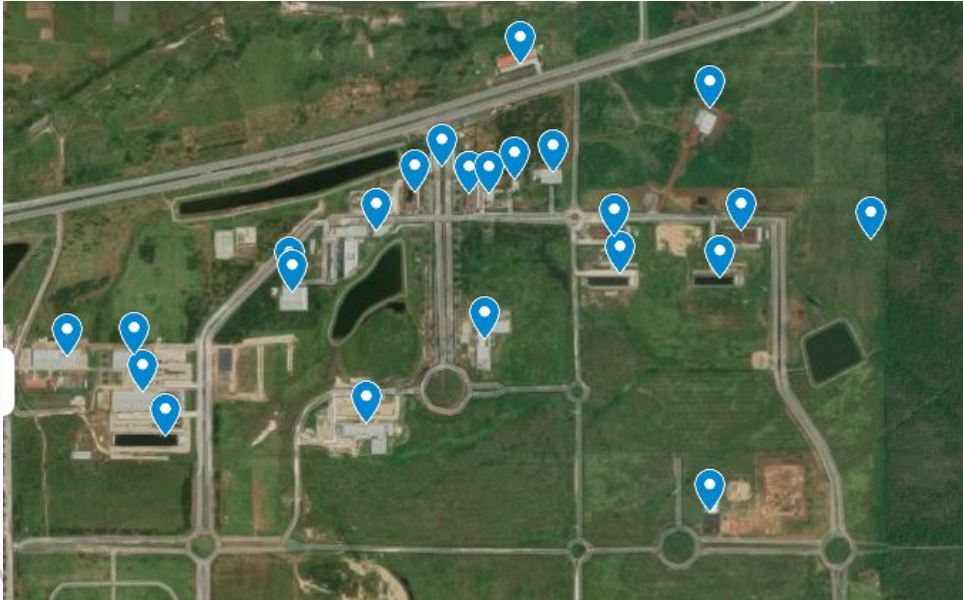


ITERA merupakan perguruan tinggi baru yang memiliki luas hampir 275 Ha, dan memiliki gedung, kebun raya, embung, taman, parkir, dan tempat lainnya. Jika seseorang hanya mempunyai waktu satu hari, dan ingin mengelilingi seluruh gedung yang ada. Pasti sangat melelahkan jika mengunjungi seluruh gedung tanpa arah dan tanpa pola. 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.





Masalah



24 Titik

275 Ha

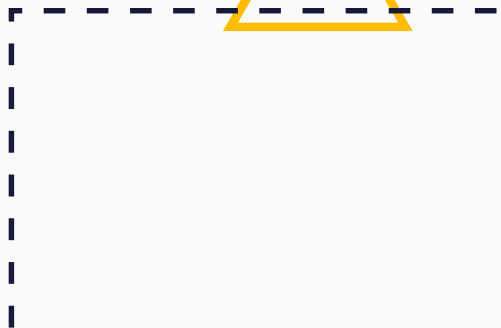
Mengelilingi
seluruhnya?





Deskripsi Data

Dengan menggunakan Google Maps kami melakukan penandaan sebanyak 24 titik yang tersebar dari gerbang, masjid, gedung perkuliahan, laboratorium, dan gedung lainnya. Setelah dilakukan didapatkan data Latitude dan Longitude masing-masing titik dan dilakukan data dari *.kml* menjadi *.csv*.





Deskripsi Data

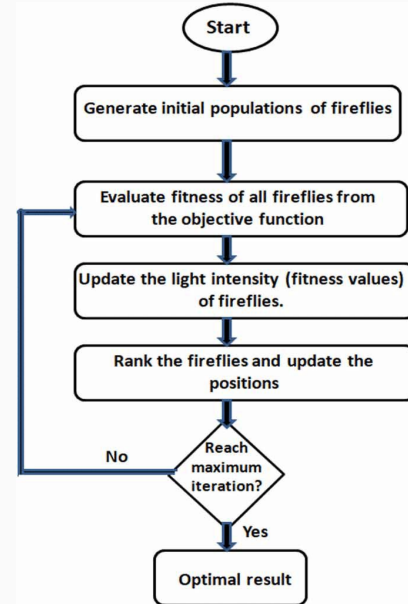
	Asrama 1	Asrama 2	Asrama 3	Asrama 4	At Tanwir Mosque	Baitul Ilmi Mosque	Galeri	Gedung A ITERA	Gedung B ITERA	Gedung C ITERA	...	Gedung Labtek 2 ITERA	Gedung Labtek 3 ITERA	Gerbang Utama	GPU 1	G' lte
Nama Gedung																
Asrama 1	0.000000	0.212562	0.179391	0.018479	0.168656	0.544826	0.213063	0.335797	0.246667	0.401103	...	0.756409	0.917457	0.291970	0.792995	0.2010
Asrama 2	0.212562	0.000000	0.039762	0.205152	0.075938	0.757345	0.425110	0.548052	0.458784	0.613644	...	0.968041	1.129832	0.503702	1.005080	0.1346
Asrama 3	0.179391	0.039762	0.000000	0.170212	0.076850	0.722972	0.392444	0.515077	0.426027	0.579937	...	0.932355	1.094956	0.471360	0.969811	0.1051
Asrama 4	0.018479	0.205152	0.170212	0.000000	0.167377	0.552772	0.223295	0.345375	0.256673	0.409808	...	0.762890	0.924915	0.302326	0.799989	0.1846
At Tanwir Mosque	0.168656	0.075938	0.076850	0.167377	0.000000	0.707505	0.373170	0.495626	0.406651	0.562672	...	0.921649	1.080464	0.450103	0.957203	0.1778

5 rows x 24 columns



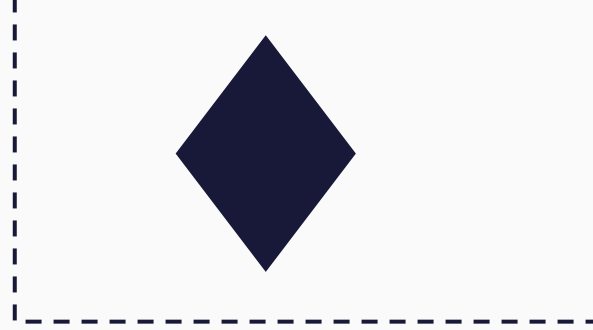
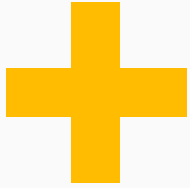
Metode

```
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
```



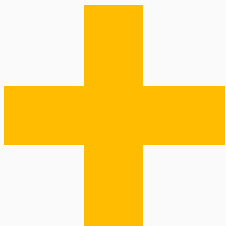
Pemodelan Individu

Inisialisasi individu merupakan individu yang di generate secara random untuk mendapatkan populasi yang dibutuhkan pada pembuatan TSP. Pada progress kali ini kami mengenerate individu dengan range 1 sampai dengan 10 secara random berdimensi 24 atau $n=24$ yang selanjutnya kami gunakan untuk mengenerate populasi.





Hasil dan Analisis





1. Solusi terbaik, parameter terbaik, dan iterasi berhenti

a. Solusi terbaik

Solusi terbaik terjadi ketika parameter $\gamma=20$. Visualisasi gambar dapat dilihat pada tabel 3.1.

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.




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



b. Parameter terbaik






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:



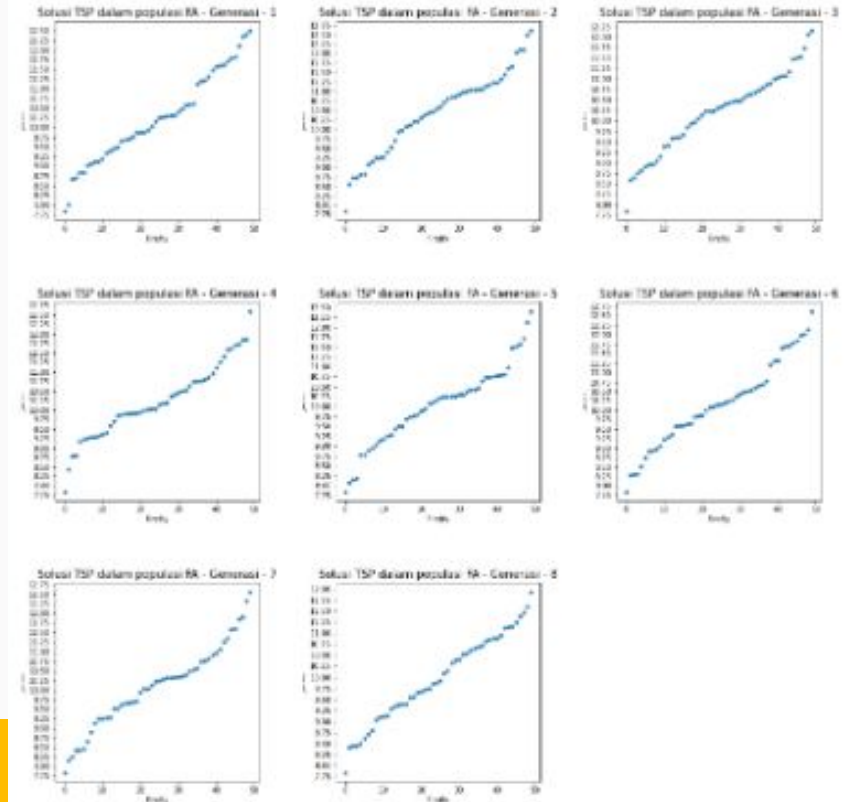
Nama Parameter	Nilai Parameter	Jarak (KM)
Alpha (α)	30	7.644359388
Beta0 (β_0)	20	7.585667705
Gamma (γ)	20	7.462324779

Tabel 3.4 Hasil Terbaik dari Parameter Metode



c. Iterasi Berhenti

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

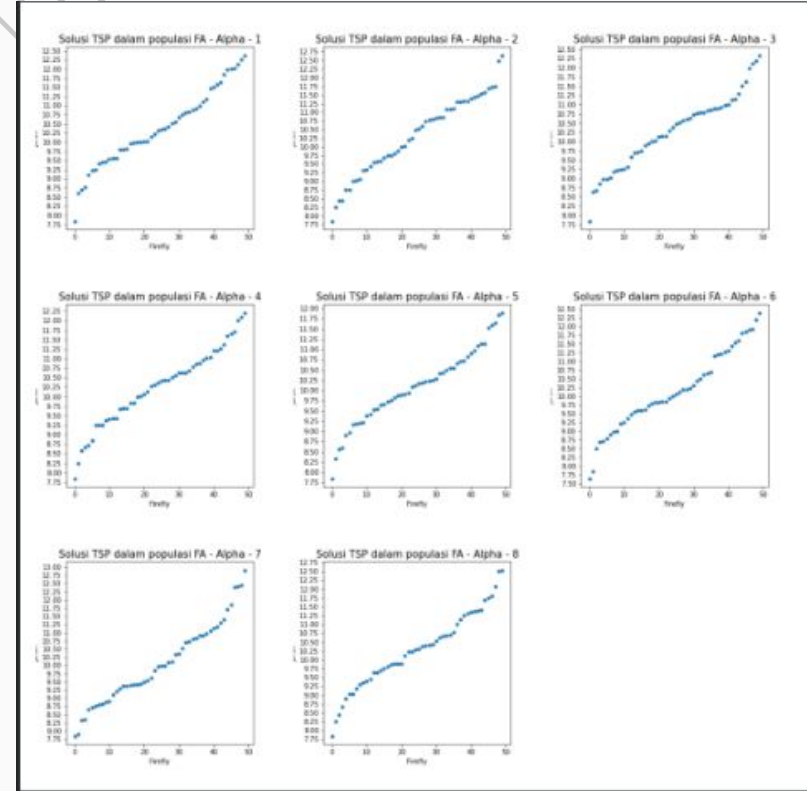


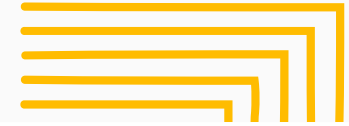
2. Visualisasi hubungan antara parameter dan solusi terbaik.



a.Parameter

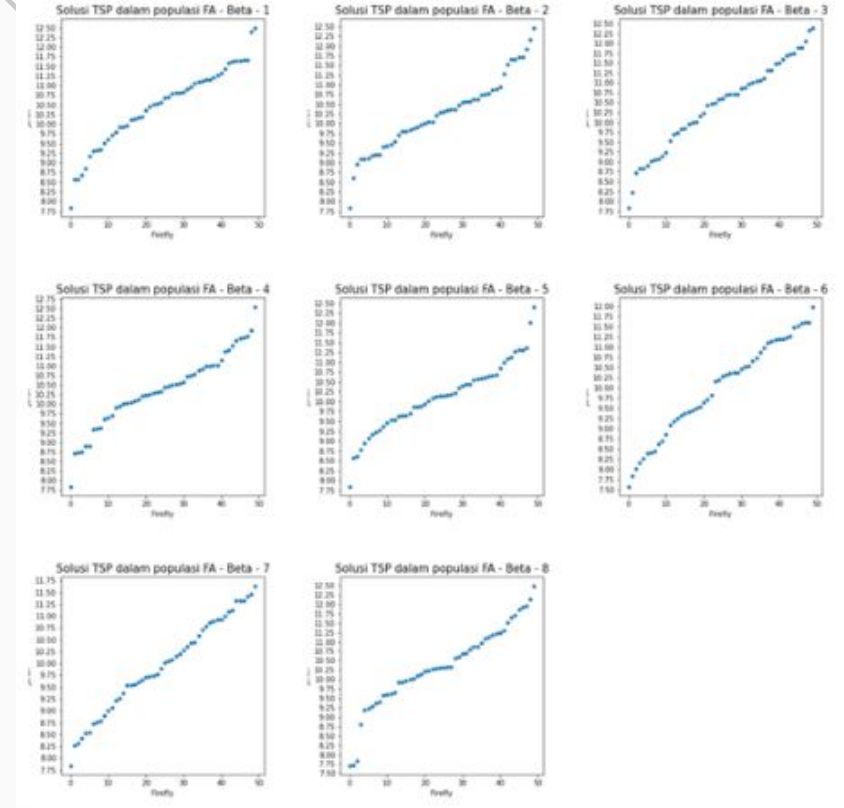
(i). Alpha (randomization rate)
Pada visualisasi diatas kami
mendapatkan solusi terbaik pada TSP
kunang – kunang dengan merubah/
memvariasikan parameter alpha
0.01,0.1,1,10,20,30,50 dan 100
didapatkankan variasi dengan solusi
terbaik pada Alpha ke – 6 dengan
nilai alpha sebesar 30 dengan jarak
7.644359.





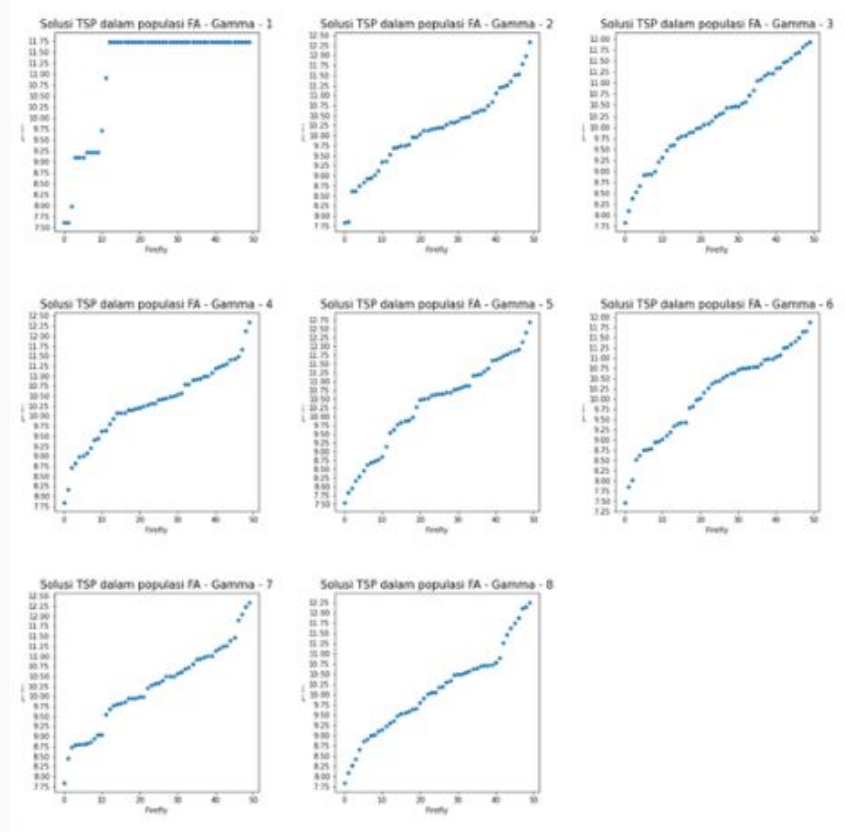
(ii). Betao (attractive firefly when
n)

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



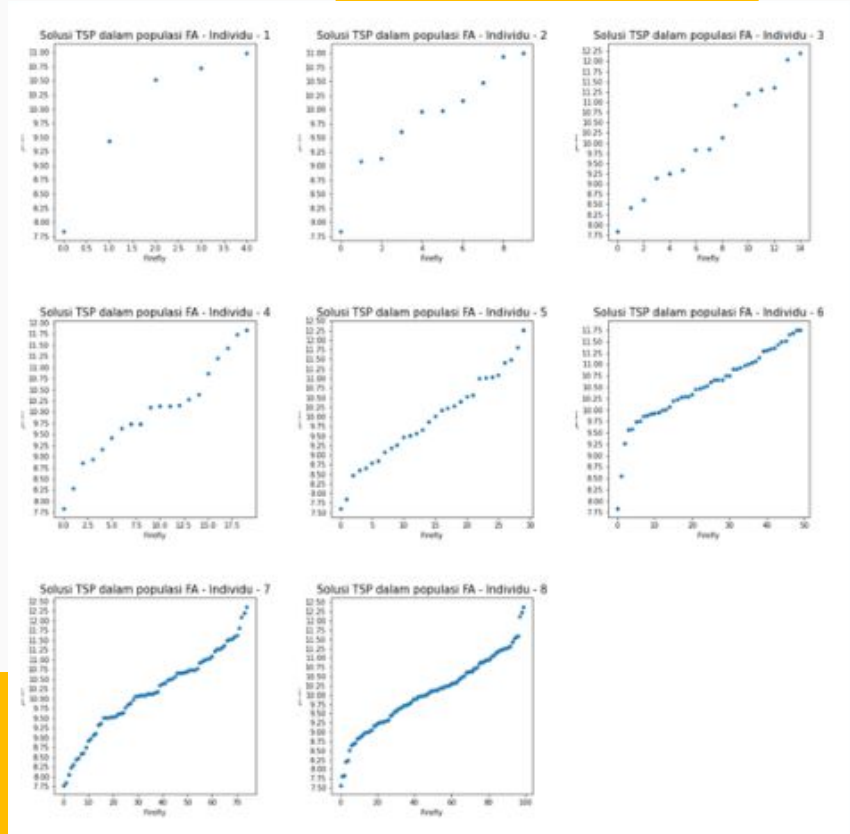


(iii). Gamma (coefficient of light)
Jika dilihat pada visualisasi diatas didapatkan solusi terbaik pada TSP kunang-kunang dengan merubah parameter Gamma 0.01, 0.1, 1, 2, 10, 20, 30, dan 50 di dapatkan variasi dengan solusi terbaik terdapat pada Gamma ke-6 dengan nilai Gamma sebesar 20.



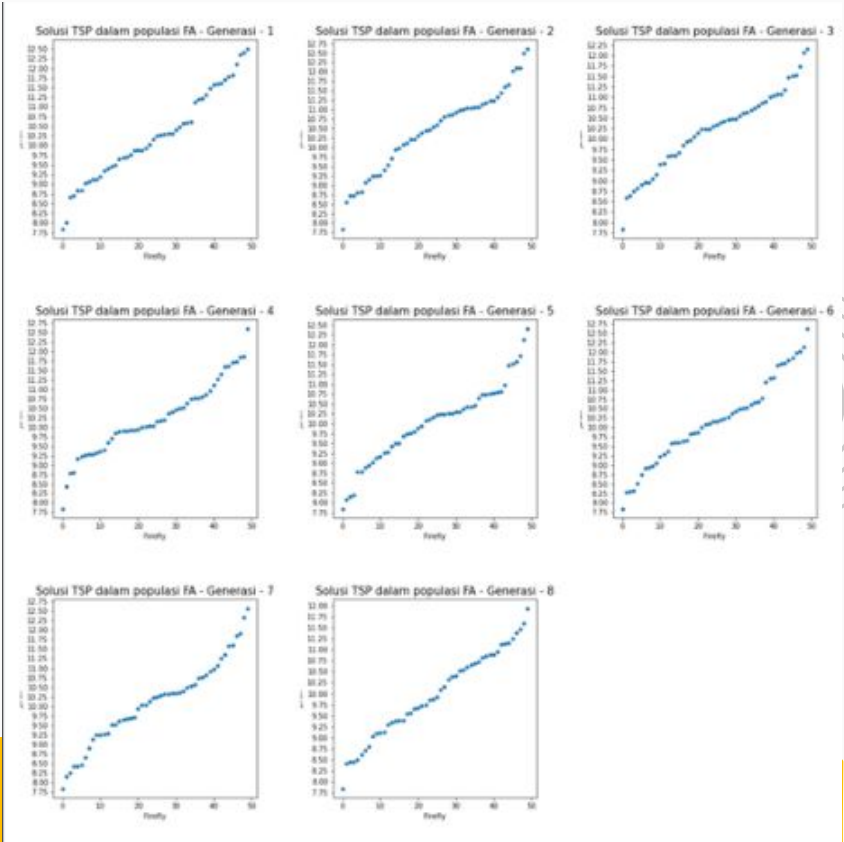
b. Jumlah Individu

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.



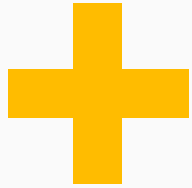
c. Maksimal Generasi

Jika dilihat pada visualisasi diatas didapatkan solusi terbaik pada TSP kunang-kunang dengan merubah parameter Jumlah maksimal generasi 1, 2, 3, 5, 10, 15, 20 dan 50 di dapatkan variasi dengan solusi terbaik terdapat pada Jumlah maksimal generasi 1-7 dengan nilai maksimal generasi sebesar 20.



Kesimpulan

1. Penerapan *Firefly Algorithm* dalam *Travelling Salesman Problem* berhasil diterapkan pada kasus Gedung Kampus Institut Teknologi Sumatera.
2. Pada percobaan, didapatkan **solusi optimal** ketika parameter nilai gamma sebesar 20 dengan jarak 7.462324779 KM.
3. Dari hasil analisis parameter r 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. Begitu pula sebaliknya.





Thanks!

