

Imam Sujiwo, 2015, *Hybrid Firefly Algorithm Dan Algoritma Simulated Annealing untuk menyelesaikan Travelling Salesman Problem*, Skripsi ini dibawah bimbingan Dr. Herry Suprajitno, M.Si dan Auli Damayanti, S,Si,M.Si. Departemen Matematika, Fakultas Sains dan Teknologi, Universitas Airlangga, Surabaya.

---

## ABSTRAK

*Traveling Salesman Problem* (TSP) adalah suatu permasalahan dalam menentukan rute yang optimal dari sejumlah rute perjalanan seorang *salesman* sehingga setiap kota hanya boleh dikunjungi tepat satu kali kemudian kembali lagi ke kota awal. Oleh karena itu, tujuan dari penulisan skripsi ini adalah menyelesaikan *Traveling Salesman Problem* dengan *hybrid Firefly Algorithm* dan algoritma *Simulated Annealing* (SA). *Firefly algorithm* terinspirasi dari perilaku kunang-kunang. Algoritma SA merupakan metode yang dianalogikan dengan proses *annealing*. *Hybrid firefly algorithm* dan algoritma SA adalah gabungan dari kedua algoritma dengan cara memproses *firefly algorithm* terlebih dahulu kemudian proses algoritma SA. Proses algoritma ini dimulai dengan inisialisasi parameter, pembangkitan solusi awal *firefly*, menghitung fungsi tujuan, menghitung intensitas cahaya, membandingkan intensitas cahaya tiap *firefly*, *update* solusi baru, menentukan *G-best*, pemilihan solusi untuk algoritma SA, melakukan proses algoritma SA terhadap solusi yang sudah dipilih dengan mutasi inversi, menyimpan solusi terbaik, dan proses berlanjut sampai maksimal iterasi. Data yang digunakan adalah data 10 kota di Jawa Timur, data 17 kota di Swiss (*Grouschield*), dan data 100 kota di Pulau Jawa. Algoritma ini diimplementasikan dengan bahasa pemrograman C++, menggunakan software Borland C++. Fungsi tujuan (jarak) minimum terbaik berdasarkan hybrid firefly algorithm dan algoritma SA didapatkan untuk data 10 kota di Timur sebesar 1022 km, untuk data 17 kota di Swiss sebesar 2039 km, sedangkan untuk data 100 kota di Pulau Jawa diperoleh jarak minimum sebesar 10149 km.

**Kata Kunci:** *Firefly Algorithm, Algoritma Simulated Annealing, Hybrid, Travelling Salesman Problem*

Imam Sujiwo, 2015, **Hybrid Firefly Algorithm and Simulated Annealing Algorithm to Solve The Travelling Salesman Problem**, This undergraduate thesis is supervised by Herry Suprajitno, M.Si and Auli Damayanti, S.Si, M.Si. Mathematics Department, Faculty of Science and Technology, Airlangga University, Surabaya.

---

## ABSTRACT

Traveling Salesman Problem is a problem of determining the optimal route from salesman's routes. Therefore, the purpose of writing this undergraduate thesis is to solve the Traveling Salesman Problem with a Hybrid firefly algorithm and Simulated Annealing Algorithm. Firefly algorithm that inspired by the behavior of fireflies. Simulated Annealing algorithm is analogous method to the annealing process. Hybrid Firefly Algorithm and Simulated Annealing are the combination of these two algorithm in a way to process firefly algorithm first then processes the SA algorithm. The process of this algorithm starts with initialization parameters, the generation of the initial population firefly, calculate the objective function, calculate the intensity of light, comparing the light intensity of each firefly, updates of new solutions, determine G- best, select solution for simulated annealing algorithm, do the simulated annealing algorithm to the selected solution with inversion mutation, memorize the best solution, find the exhausted solution and the process running until maximum iteration. The data used sois the data 10 cities in East Java, the data 17 cities in Swiss, and 100 cities in Java Island. This algorithm was being simulated with C++ programming language, using Borland C++ software. The objective function (distance) minimum based on hybrid firefly algorithm an simulated annealing for data 10 cities in East Java is 1022 km, for data 17 cities in Swiss is 2039 km, while the data for 100 cities in Java Island obtained a minimum distance is 10149 km..

**Keywords:** Firefly Algorithm, Simulated Annealing Algorithm, Hybrid, Travelling Salesman Problem