

ABSTRAK

Optimisasi merupakan proses untuk memperoleh solusi terbaik dari sejumlah alternatif dengan mempertimbangkan tujuan dan batasan yang ada. Salah satu permasalahan optimisasi yang banyak diteliti adalah *Traveling Salesman Problem* (TSP), yaitu permasalahan pencarian rute terpendek yang mengunjungi setiap kota tepat satu kali dan kembali ke kota asal. Penelitian ini mengimplementasikan algoritma *Dynamic Programming* dengan pendekatan Held-Karp untuk menyelesaikan TSP. Pengujian dilakukan dengan jumlah kota yang ditingkatkan secara bertahap mulai dari 3 kota. Pengujian dihentikan pada 24 kota karena terjadinya *combinatorial explosion* pada jumlah *state* yang harus dievaluasi, sehingga mencapai batas kemampuan komputasi perangkat keras yang digunakan. Pada pengujian dengan 24 kota, waktu komputasi mencapai 13.142,23 detik (sekitar 3,65 jam). Hasil penelitian menunjukkan bahwa algoritma Held-Karp mampu menghasilkan solusi optimal pada seluruh skenario pengujian. Selain itu, implementasi *Dynamic Programming* dengan mekanisme *memoization* berhasil menghindari perhitungan ulang terhadap *state* yang sama sehingga solusi optimal dapat diperoleh secara sistematis meskipun jumlah *state* meningkat secara eksponensial. Berdasarkan hasil tersebut, algoritma *Held-Karp* efektif digunakan untuk TSP pada skala relatif sedikit hingga menengah (hingga 24 kota), namun menjadi kurang praktis untuk jumlah kota yang lebih besar akibat meningkatnya kebutuhan komputasi secara eksponensial.

Kata Kunci: *Dynamic Programming, Held-Karp, Traveling Salesman Problem*

ABSTRACT

Optimization is the process of obtaining the best solution from a set of alternatives by considering existing objectives and constraints. One of the most widely studied optimization problems is the *Traveling Salesman Problem* (TSP), which aims to determine the shortest route that visits each city exactly once and returns to the starting city. This research implements the *Dynamic Programming* algorithm using the Held-Karp approach to solve the TSP. The experiments were conducted by gradually increasing the number of cities, starting from 3 cities. The experiments were terminated at 24 cities due to the occurrence of a *combinatorial explosion* in the number of states that had to be evaluated, reaching the computational limit of the hardware used in this study. In the largest experiment involving 24 cities, the computation time reached 13,142.23 seconds (approximately 3.65 hours). The results show that the Held-Karp algorithm successfully produced optimal solutions for all test scenarios. Furthermore, the implementation of *Dynamic Programming* with the *memoization* mechanism effectively avoided redundant state computations, enabling optimal solutions to be obtained systematically despite the exponential growth in the number of states. Based on these findings, the Held-Karp algorithm is effective for solving TSP instances of small to medium size (up to 24 cities), but becomes impractical for larger problem instances due to the exponential growth in computational requirements.

Keywords: Dynamic Programming, Held-Karp, Traveling Salesman Problem