



# Artificial Intelligence

Practicum Module

Week 7:

## K-Nearest Neighbors

**Prepared by:**

Assistant Lecturer of Artificial Intelligence 2024

Department of Computer Science

University of Lampung

# **Authoring Team**

## **Lecturer**

Prof. Admi Syarif, PhD  
Rahman Taufik, M.Kom.

## **Authors**

- Adli Fiqrullah
- Alfa Rizki Fadlillah
- Auvar Mahsa Fahlevi
- Egy Vedriyanto
- John Wilken Christoper Nainggolan
- Kartika Sari
- M. Raditya Adhirajasa

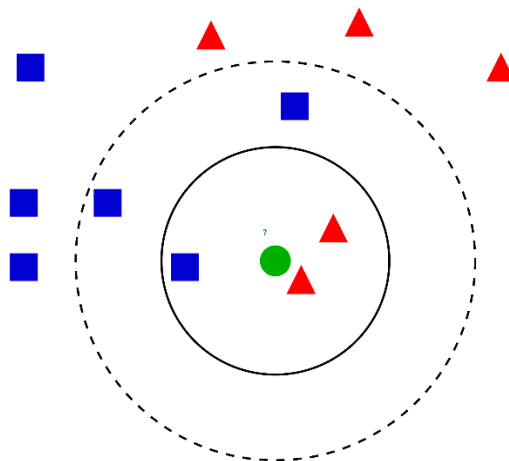
## Deskripsi Singkat

Algoritma K-Nearest Neighbors (KNN) adalah teknik pembelajaran mesin yang populer digunakan untuk tugas klasifikasi dan regresi. Algoritma ini bergantung pada gagasan bahwa titik data yang serupa cenderung memiliki label atau nilai yang serupa. Selama fase pelatihan, algoritma KNN menyimpan seluruh kumpulan data pelatihan sebagai referensi. Ketika membuat prediksi, algoritma ini menghitung jarak antara titik data input dan semua titik data contoh pelatihan, menggunakan metrik jarak yang dipilih seperti Euclidean Distance.

## Capaian Pembelajaran

1. Memberikan gambaran umum algoritma K-Nearest Neighbors
2. Memahami konsep algoritma K-Nearest Neighbors

## Materi



Algoritma K-Nearest Neighbors (k-NN) adalah salah satu algoritma yang paling sederhana dalam pembelajaran mesin. Meskipun sederhana, algoritma ini sering kali sangat efektif dalam berbagai konteks, terutama dalam klasifikasi dan regresi. Algoritma K-NN mengukur kesamaan antara objek-objek dalam ruang fitur dengan menghitung jarak antara mereka. Jarak ini dapat dihitung menggunakan berbagai metrik, seperti jarak Euclidean, Manhattan, atau Mahalanobis, tergantung pada sifat data dan kebutuhan masalah.

Ketika kita memiliki objek yang akan diprediksi, algoritma k-NN mencari k objek terdekat atau "tetangga terdekat" dari objek tersebut dalam ruang fitur. K adalah parameter yang harus ditentukan sebelumnya dan biasanya merupakan bilangan bulat positif.

Nilai  $k$  yang lebih rendah dapat memiliki varians yang tinggi, tetapi bias yang rendah. Sedangkan nilai  $k$  yang lebih besar dapat menyebabkan bias yang tinggi dan varians yang lebih rendah. Pilihan  $k$  akan sangat bergantung pada data input karena data dengan lebih banyak outlier atau noise kemungkinan akan berkinerja lebih baik dengan nilai  $k$  yang lebih tinggi. Secara keseluruhan, disarankan untuk memilih nilai  $k$  berupa angka ganjil untuk menghindari ikatan dalam klasifikasi. Strategi cross validation juga dapat digunakan untuk membantu kita memilih  $k$  yang optimal untuk dataset yang kita miliki.

Langkah-langkah algoritma KNN adalah sebagai berikut:

1. Memilih Nilai  $k$ : Langkah pertama adalah memilih jumlah tetangga terdekat ( $k$ ) yang akan digunakan dalam proses klasifikasi atau regresi.
2. Menghitung Jarak: Setelah nilai  $k$  ditentukan, langkah berikutnya adalah menghitung jarak antara objek yang akan diprediksi dengan setiap objek dalam data latih. Jarak ini dapat dihitung menggunakan berbagai metrik, seperti jarak Euclidean, Manhattan, atau Mahalanobis, tergantung pada sifat data dan masalah yang dihadapi.
3. Menentukan Tetangga Terdekat: Setelah jarak dihitung, langkah selanjutnya adalah menemukan  $k$  objek terdekat dari objek yang akan diprediksi.
4. Mengambil Mayoritas Label: untuk klasifikasi, kita mengambil mayoritas label dari  $k$  tetangga terdekat tersebut. Sedangkan untuk regresi, kita dapat mengambil rata-rata nilai dari  $k$  tetangga terdekat.

Algoritma  $k$ -NN memiliki beragam aplikasi dalam berbagai bidang, termasuk klasifikasi pola, pengenalan pola, sistem rekomendasi. Kelebihan utamanya adalah kemudahan implementasi dan interpretasi, serta kemampuannya untuk menangani dataset yang besar dan kompleks tanpa memerlukan asumsi yang kuat tentang distribusi data.