# Laporan Kuliah Praktisi Analisis Algoritma

## Kompleksitas Pada Machine Learning

1. Paradigma Komputasi

* Terdapat dua paradigma komputasi: hard computing dan soft computing.
* ML termasuk soft computing, di mana solusi menjadi lebih baik seiring dengan waktu yang dihabiskan.

1. Kompleksitas Waktu dan Ruang

* Waktu (Time) dan Ruang (Space) Complexity: Dua dimensi penting yang menentukan seberapa efisien suatu algoritma ML.
* Faktor-faktor Pengaruh: Jumlah sampel pelatihan, fitur, struktur model (seperti pohon keputusan atau neuron pada jaringan saraf), dan parameter lainnya.

1. Kompleksitas Berdasarkan Jenis Algoritma

* Supervised Learning:
* Kompleksitas waktu tergantung pada sejumlah besar data berlabel.
* Ruang kompleksitas dipengaruhi oleh struktur model yang digunakan.
* Unsupervised Learning:
* Kompleksitas tergantung pada kemampuan algoritma untuk menemukan pola dalam data tanpa bimbingan.
* Algoritma seperti K-Means Clustering memiliki kompleksitas yang berkaitan dengan jumlah cluster.
* Deep Learning (Neural Networks):
* Kompleksitas terkait dengan jumlah lapisan dan neuron dalam jaringan.
* Waktu dan ruang complexity untuk pelatihan sangat bergantung pada ukuran dataset dan arsitektur jaringan.

1. Kompleksitas Algoritma Spesifik

* K-Nearest Neighbour (KNN):
* Prediction phase: Bergantung pada implementasi algoritma, dengan KNN-KD Tree Algorithm memiliki kompleksitas O(np) dan KNN-Brute Force Algorithm memiliki kompleksitas O(np^2).
* Training phase: Bergantung pada algoritma pembangunan pohon KD dan memiliki kompleksitas O(np log(n)).
* Decision Tree:
* Training phase: Bergantung pada algoritma pembentukan struktur pohon keputusan, dengan kompleksitas O(n^2p).
* Prediction phase: Dalam situasi terburuk, memiliki kompleksitas O(p).
* Random Forest:
* Training phase: Bergantung pada pembangunan pohon keputusan individu dan jumlah pohon, dengan kompleksitas O(n^2p ntrees).
* Prediction phase: Dalam situasi terburuk, memiliki kompleksitas O(p ntrees).
* Logistic Regression:
* Training phase: Bergantung pada penemuan vektor bobot dalam algoritma regresi logistik, dengan kompleksitas O(p^2n + p^3).
* Prediction phase: Dalam situasi terburuk, memiliki kompleksitas O(p).
* Support Vector Machine (Kernel):
* Training phase: Bergantung pada pembangunan model SVM dengan kernel, dengan kompleksitas O(n^2p + n^3).
* Prediction phase: Bergantung pada jumlah vektor dukungan dan jumlah fitur, dengan kompleksitas O(nsvp).
* Neural Networks:
* Training phase: Bergantung pada arsitektur jaringan dan perhitungan bobot pra-terlatih, dengan kompleksitas bervariasi.
* Prediction phase: Bergantung pada jumlah lapisan dan neuron, dengan kompleksitas O(pnl1 + nl1nl2 + ...).
* K-Means Clustering:
* Training phase: Bergantung pada pembentukan cluster, dengan kompleksitas O(nkl).
* Prediction phase: Bergantung pada jumlah cluster dan fitur, dengan kompleksitas O(npk).

## Kompleksitas Komputasi Pada Sudut Pandang Praktis

**Trade-off antara Kompleksitas dan Generalisasi**

* 1. Overfitting dan Kompleksitas:
* Model kompleks mungkin memiliki lebih banyak parameter, fitur, atau lapisan, memungkinkannya untuk belajar dari data dan membuat prediksi yang akurat.
* Model yang terlalu kompleks dapat dengan mudah menghafal detail dari set pelatihan, menciptakan model yang tidak dapat menangani variasi atau data baru.
* Ini dapat menyebabkan performa model yang buruk pada data yang belum pernah dilihat sebelumnya.

1. Underfitting dan Kesederhanaan:

* Model sederhana mungkin memiliki lebih sedikit parameter, fitur, atau lapisan, membuatnya lebih mudah dilatih dan diinterpretasi.
* Model yang terlalu sederhana mungkin tidak memiliki kemampuan untuk menangkap nuansa atau pola kompleks dalam data pelatihan.
* Kesederhanaan dapat menyebabkan kurangnya adaptasi terhadap variasi dalam data.

## Contoh Kasus (selain yang ada dalam kuliah praktisi)