Pengantar MACHINE LEARNING

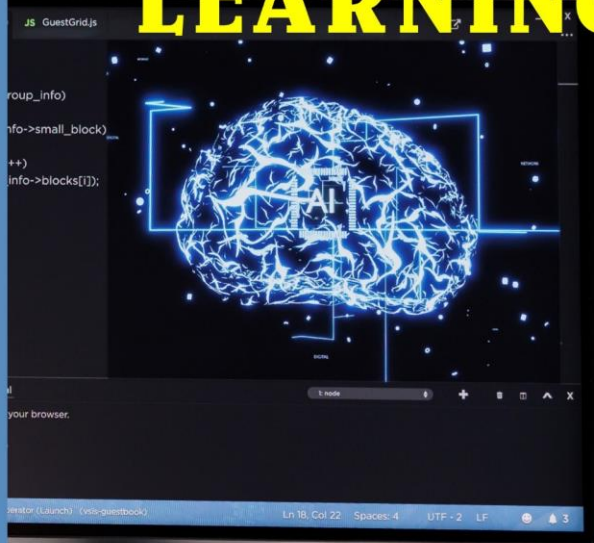
Machine Learning (ML) merupakan salah satu cabang dari kecerdasan buatan (Artificial Intelligence) yang berfokus pada pengembangan algoritma dan model yang memungkinkan komputer untuk "belajar" dari data. Alih-alih diprogram secara eksplisit untuk melakukan tugas tertentu, komputer dilatih untuk mengenali pola dan membuat keputusan berdasarkan data yang mereka analisis. Di era digital saat ini, machine learning telah menjadi salah satu bidang yang paling cepat berkembang di dunia teknologi. Dari pengenalan pola sederhana hingga sistem pembelajaran mendalam (deep learning) yang kompleks, machine learning telah menjadi fondasi dari banyak inovasi teknologi modern, seperti pengenalan wajah, kendaraan otonom, asisten virtual, dan sistem rekomendasi. Buku ini membahas tentang Konsep Dasar Machine Learning, Supervised Learning, Regresi Linear, K-mean Clustering, K-medoid Clustering.

PT MAFY MEDIA LITERASI INDONESIA
ANGGOTA IKAPI 041/SBA/2023
Email : penerbitmafy@gmail.com
Website : penerbitmafy.com
FB : Penerbit Mafy

PENDIDIKAN
ISBN 978-623-8758-78-4
9 786238 758784

PENGANTAR MACHINE LEARNING

Pengantar MACHINE LEARNING



Panji Bintoro, Ratnasari, Edy Wihardjo,
Indah Pratiwi Putri, Andi Asari

Pengantar
Machine Learning

UU No 28 Tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

i. penggunaan kutipan singkat ciptaan dan/atau produk hak terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
ii. penggandaan ciptaan dan/atau produk hak terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
iii. penggandaan ciptaan dan/atau produk hak terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan fonogram yang telah dilakukan pengumuman sebagai bahan ajar; dan
iv. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu ciptaan dan/atau produk hak terkait dapat digunakan tanpa izin pelaku pertunjukan, produser fonogram, atau lembaga penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap orang yang dengan tanpa hak dan/atau tanpa izin pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

Pengantar *Machine Learning*

Panji Bintoro
Ratnasari
Edy Wihardjo
Indah Pratiwi Putri
Andi Asari

MAFY media literasi INDONESIA

PENGANTAR MACHINE LEARNING

Penulis:
Panji Bintoro, Ratnasari, Edy Wihardjo, Indah Pratiwi Putri, Andi Asari

Editor:
Andi Asari

Desainer:
Tim Mafy

Sumber Gambar Cover:
www.freepik.com

Ukuran:
viii, 127 hlm, 15,5 cm x 23 cm

ISBN:
978-623-8758-78-4

Cetakan Pertama:
September 2024

PT MAFY MEDIA LITERASI INDONESIA
ANGGOTA IKAPI 041/SBA/2023
Kota Solok, Sumatera Barat, Kode Pos 27312
Kontak: 081374311814
Website: www.penerbitmafy.com
E-mail: penerbitmafy@gmail.com

Prakata

Segala puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas pertolongan dan limpahan rahmat-Nya sehingga penulis bisa menyelesaikan buku yang berjudul **Pengantar Machine Learning** ini disusun secara lengkap dengan tujuan untuk memudahkan para pembaca memahami isi buku ini. Buku ini membahas tentang Konsep Dasar Machine Learning, Supervised Learning, Regresi Linear, K-mean Clustering, K-medoid Clustering.

Kami menyadari bahwa buku yang ada di tangan pembaca ini masih banyak kekurangan. Maka dari itu kami sangat mengharapkan saran untuk perbaikan buku ini di masa yang akan datang. Dan tidak lupa kami mengucapkan terima kasih kepada semua pihak yang telah membantu dalam proses penerbitan buku ini. Semoga buku ini dapat membawa manfaat dan dampak positif bagi para pembaca.

Penulis, Malang 15 September 2024

Daftar Isi

BAB 1

Konsep Dasar Machine Learning

-Panji Bintoro-

A. Pengantar *Machine Learning*

Machine Learning (ML) merupakan salah satu cabang dari kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada pengembangan algoritma dan model yang memungkinkan komputer untuk ”belajar” dari data. Alih-alih diprogram secara eksplisit untuk melakukan tugas tertentu, komputer dilatih untuk mengenali pola dan membuat keputusan berdasarkan data yang mereka analisis.

1. Definisi *Machine Learning*

Secara umum, *machine learning* dapat didefinisikan sebagai suatu bidang studi yang memberikan kemampuan kepada sistem komputer untuk belajar dan meningkatkan kinerjanya dari pengalaman tanpa harus diprogram secara eksplisit untuk tugas tertentu (goodfellow, bengio and courville, 2016). Arthur Samuel, seorang pionir dibidang ini, mendefinisikan *machine*

learning sebagai "bidang studi yang memberi komputer kemampuan untuk belajar tanpa diprogram secara eksplisit." Definisi ini menggambarkan esensi dari *Machine Learning*, yaitu kemampuan adaptif sistem dalam meningkatkan performa berdasarkan data yang terus berkembang.

2. Sejarah dan Perkembangan *Machine Learning*

Machine learning bukanlah konsep yang baru. Sejak akhir tahun 1950-an, para peneliti telah berupaya mengembangkan algoritma yang memungkinkan mesin untuk belajar dari data. Salah satu pionirnya adalah Arthur Samuel, yang mengembangkan program bermain catur yang dapat "belajar" dari permainan sebelumnya untuk menjadi lebih baik. Seiring waktu, *machine learning* berkembang pesat dengan meningkatnya kemampuan komputasi dan ketersediaan data dalam jumlah besar (murphy, 2012).

Di era digital saat ini, *machine learning* telah menjadi salah satu bidang yang paling cepat berkembang di dunia teknologi. Dari pengenalan pola sederhana hingga sistem pembelajaran mendalam (deep learning) yang kompleks, *machine learning* telah menjadi fondasi dari banyak inovasi teknologi modern, seperti pengenalan wajah, kendaraan otonom, asisten virtual, dan sistem rekomendasi.

3. Peran *Machine Learning* dalam Kecerdasan Buatan (AI)

Machine learning adalah salah satu cabang utama dari Kecerdasan Buatan (AI). Jika AI adalah tujuan akhir untuk menciptakan mesin yang cerdas dan mampu meniru perilaku manusia, maka *machine learning* adalah salah satu jalan menuju tujuan tersebut. Dengan ML,

komputer dapat mempelajari pola dari data yang ada, membuat prediksi, dan mengambil keputusan tanpa memerlukan program yang dirancang khusus untuk setiap tugas. Ini memungkinkan AI untuk berkembang lebih cepat dan beradaptasi dengan berbagai jenis masalah.

Contoh klasik dari penerapan *machine learning* dalam AI adalah sistem pengenalan gambar. Dalam sistem ini, model ML dilatih menggunakan ribuan atau jutaan gambar yang telah diberi label untuk mengenali objek tertentu, seperti wajah manusia atau hewan. Setelah dilatih, model dapat mengenali objek yang serupa dalam gambar baru dengan tingkat akurasi yang tinggi.

4. Penerapan *Machine Learning* dalam Kehidupan Sehari-hari

Machine learning telah menjadi bagian integral dari kehidupan sehari-hari kita, meskipun seringkali kita tidak menyadarinya. Beberapa contoh umum penerapan ML adalah:

a. Sistem Rekomendasi: Platform seperti Netflix, YouTube, dan Spotify menggunakan *machine learning* untuk merekomendasikan konten berdasarkan preferensi dan kebiasaan pengguna.
b. Pengenalan Suara: Asisten virtual seperti Siri, Google Assistant, dan Alexa menggunakan teknologi ML untuk mengenali dan memahami perintah suara pengguna.
c. Deteksi Penipuan: Bank dan perusahaan kartu kredit menggunakan ML untuk mendeteksi transaksi yang mencurigakan dan mencegah penipuan.

d. Pencarian di Internet: Mesin pencari seperti Google menggunakan ML untuk memahami apa yang dicari pengguna dan menyajikan hasil yang paling relevan.

Dengan semakin banyaknya data yang dihasilkan setiap hari, peran *machine learning* dalam mengolah dan memanfaatkan data ini menjadi semakin penting. ML tidak hanya memungkinkan sistem untuk membuat keputusan yang lebih baik, tetapi juga memungkinkan prediksi yang lebih akurat tentang tren masa depan, memberikan keuntungan kompetitif bagi organisasi yang menggunakannya. Bagian ini memberikan pengantar yang jelas tentang apa itu *machine learning*, sejarah dan perkembangannya, serta penerapannya dalam berbagai aspek kehidupan sehari-hari. Ini membantu pembaca memahami dasar-dasar sebelum melanjutkan ke topik yang lebih mendalam.

B. Komponen Utama *Machine Learning*

Machine Learning (ML) adalah bidang yang kompleks dan terdiri dari beberapa komponen kunci yang saling berinteraksi untuk menciptakan model yang efektif. Memahami komponen utama ini sangat penting bagi siapa saja yang ingin mendalami atau menerapkan *machine learning*. Pada bagian ini, kita akan membahas empat komponen utama: data, algoritma, model, dan evaluasi.

1. Jenis dan Sumber Data

Data adalah fondasi dari *machine learning*. Tanpa data yang memadai, sebuah model ML tidak dapat belajar dan membuat prediksi yang akurat. Data dalam ML dapat berupa angka, teks, gambar, suara, dan banyak

lagi, tergantung pada jenis masalah yang ingin diselesaikan (bishop, 2006).

a. Jenis Data
    1) Data Terstruktur merupakan data yang terorganisir dalam format tabel, seperti spreadsheet atau database relasional. Contoh: catatan keuangan, data pelanggan.
    2) Data Tidak Terstruktur merupakan data yang tidak memiliki format tertentu, seperti teks bebas, gambar, atau video. Contoh: tweet, artikel berita, foto.
    3) Data Semi Terstruktur merupakan data yang memiliki elemen-elemen yang terstruktur tetapi tidak sepenuhnya sesuai dengan skema tabel. Contoh: XML, JSON.

b. Sumber Data
    1) Data Internal merupakan data yang dikumpulkan oleh organisasi dari aktivitas bisnisnya sendiri, seperti data penjualan, data pelanggan, dan lain-lain.
    2) Data Eksternal merupakan data yang diperoleh dari luar organisasi, seperti data pasar, data media sosial, dan data dari pihak ketiga.
    3) Data yang Dihasilkan merupakan yang dihasilkan secara khusus untuk tujuan tertentu, seperti melalui eksperimen atau survei.

Kualitas dan kuantitas data sangat menentukan keberhasilan sebuah model ML. Oleh karena itu, pengumpulan, pembersihan, dan pra-pemrosesan data

adalah langkah-langkah penting dalam proses *machine learning*.

2. Algoritma Dasar *Machine Learning*

Algoritma adalah jantung dari *machine learning*. Algoritma adalah serangkaian instruksi atau prosedur matematis yang digunakan untuk menemukan pola dalam data dan membangun model yang dapat membuat prediksi atau keputusan.

a. Algoritma *Supervised Learning* merupakan salah satu jenis pembelajaran mesin (*machine learning*) di mana model dilatih menggunakan dataset yang berisi *input* (fitur) dan *output* yang diharapkan (label).
b. Algoritma *Unsupervised Learning* merupakan jenis pembelajaran mesin (*machine learning*) di mana model dilatih menggunakan dataset yang hanya berisi *input* tanpa label atau *output* yang diharapkan.
c. Algoritma *Semi-Supervised Learning* merupakan pendekatan dalam pembelajaran mesin yang menggabungkan elemen dari *supervised learning* dan *unsupervised learning*.
d. Algoritma *Reinforcement Learning* merupakan salah satu jenis pembelajaran mesin di mana agen (*agent*) belajar untuk membuat keputusan dengan cara berinteraksi dengan lingkungan dan menerima umpan balik berupa hadiah (*rewards*) atau hukuman (*penalties*).

Pemilihan algoritma yang tepat bergantung pada jenis data yang tersedia dan masalah spesifik yang ingin diselesaikan.

3. Pengertian dan Penggunaan Model

Model dalam *machine learning* adalah representasi matematis dari hubungan yang dipelajari oleh algoritma dari data. Model adalah hasil akhir dari proses pelatihan dan digunakan untuk membuat prediksi atau mengambil keputusan berdasarkan data baru (hastie, tibshirani *and* friedman, 2009).

a. *Training* model yaitu proses di mana algoritma belajar dari data pelatihan untuk menemukan pola dan parameter terbaik yang menggambarkan hubungan dalam data.
b. *Testing* model yaitu proses di mana Setelah model dilatih, model tersebut diuji pada data yang tidak pernah dilihat sebelumnya (data uji) untuk meng-evaluasi kinerjanya.
c. *Overfitting* dan *Underfitting*
    1) *Overfitting* terjadi ketika model terlalu rumit dan menangkap noise atau detail yang tidak relevan dalam data pelatihan, sehingga kinerjanya buruk pada data baru.
    2) *Underfitting* terjadi ketika model terlalu sederhana dan gagal menangkap pola yang sebenarnya ada dalam data.

Tujuan dari pembangunan model adalah untuk menemukan keseimbangan antara akurasi dan kompleksitas, agar model mampu menggeneralisasi dengan baik pada data baru.

4. Evaluasi dan Pengukuran Kinerja Model

Evaluasi adalah tahap penting dalam *machine learning*, karena membantu mengukur seberapa baik

model bekerja dan apakah perlu dilakukan perbaikan atau penyesuaian.

a. Metrik Evaluasi
   1) Akurasi merupakan persentase prediksi yang benar dari total prediksi yang dibuat oleh model. Umumnya digunakan untuk klasifikasi biner.
   2) *Precision*, *Recall*, dan F1-*score* merupakan Metrik yang lebih rinci untuk mengevaluasi model klasifikasi, terutama ketika terdapat ketidakseimbangan kelas.
   3) *Mean Squad Error* (MSE) dan *Root Mean Squad Error* (RMSE) digunakan untuk mengukur kinerja model regresi, menunjukkan seberapa jauh prediksi model dari nilai sebenarnya.
   4) *Area Under Curve* (AUC)-ROC digunakan untuk mengukur kemampuan model dalam membedakan antara kelas positif dan negatif.

b. *Cross Validation* merupakan teknik evaluasi yang membagi data menjadi beberapa subset, melatih model pada beberapa subset dan mengujinya pada yang lain. Ini membantu dalam mengevaluasi kinerja model secara lebih andal dengan meminimalkan bias yang mungkin timbul dari pembagian data yang tidak representatif.

Evaluasi model membantu dalam memastikan bahwa model tidak hanya bekerja baik pada data pelatihan, tetapi juga dapat menggeneralisasi dengan baik pada data baru. Bagian ini menjelaskan komponen-komponen utama yang membentuk dasar dari *machine learning*, memberikan pemahaman yang mendalam

tentang bagaimana data, algoritma, model, dan evaluasi bekerja bersama untuk menciptakan solusi yang efektif dalam berbagai aplikasi ML.

C. Kategori *Machine Learning*

Machine Learning (ML) adalah bidang yang luas dan beragam, dengan berbagai pendekatan yang dapat digunakan untuk mengatasi berbagai jenis masalah. Berdasarkan cara data digunakan dan bagaimana model belajar, *machine learning* dapat dibagi menjadi beberapa kategori utama: *Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, dan Reinforcement Learning*. Masing-masing kategori ini memiliki karakteristik, metode, dan aplikasi yang berbeda.

1. *Supervised Learning*

Supervised learning adalah salah satu metode paling umum dalam *machine learning*. Dalam pendekatan ini, model dilatih menggunakan data yang telah diberi label, yang berarti bahwa setiap contoh data pelatihan disertai dengan output atau hasil yang benar (russell *and* norvig, 2020).

a. Proses: Model belajar dengan mencocokkan input dengan output yang sesuai. Ini dilakukan dengan meminimalkan kesalahan antara prediksi model dan *output* yang benar melalui teknik optimasi. Selama proses pelatihan, model menyesuaikan parameternya untuk meminimalkan perbedaan antara prediksi dan nilai sebenarnya.

b. Contoh Algoritma

1) *Regresi Linear*: Digunakan untuk memprediksi nilai numerik yang kontinu, seperti harga rumah atau suhu.

2) *Decision Tree*: Digunakan untuk klasifikasi atau regresi, dengan membagi dataset menjadi subset yang lebih kecil berdasarkan fitur yang paling informatif.
3) *Support Vector Machine* (SVM): Digunakan untuk klasifikasi dengan memisahkan data menggunakan hyperplane yang memaksimalkan margin antara dua kelas.
4) *Neural Networks*: Digunakan untuk berbagai aplikasi, dari klasifikasi gambar hingga pengenalan suara, terutama dalam bentuk *deep learning*.

c. Aplikasi
1) Klasifikasi: Seperti pengenalan tulisan tangan (*digit recognition*) atau deteksi spam pada email.
2) Regresi: Seperti prediksi harga saham atau penjualan.

Supervised learning sangat efektif ketika data pelatihan yang berlabel tersedia dalam jumlah yang cukup dan representatif terhadap masalah yang ingin diselesaikan.

2. *Unsupervised Learning*

Unsupervised learning digunakan ketika data yang tersedia tidak memiliki label, yang berarti tidak ada output yang benar atau hasil yang diketahui. Tujuan dari *unsupervised learning* adalah untuk menemukan pola atau struktur tersembunyi dalam data.

a. Proses: Model belajar untuk mengidentifikasi pola atau kelompok dalam data tanpa pengawasan atau bimbingan dari *output* yang diketahui. Ini sering

melibatkan teknik-teknik seperti clustering atau reduksi dimensi.

b. Contoh Algoritma
   1) *K-Means Clustering*: Digunakan untuk membagi data ke dalam kelompok-kelompok (*clusters*) berdasarkan kemiripan fitur.
   2) *Hierarchical Clustering*: Membuat pohon hierarki dari cluster yang dapat digunakan untuk memahami hubungan antar kelompok data.
   3) *Principal Component Analysis* (PCA): Digunakan untuk mengurangi dimensi data dengan mengidentifikasi komponen-komponen utama yang menjelaskan sebagian besar variasi dalam data.
   4) *Autoencoders*: Neural networks yang digunakan untuk belajar representasi data yang efisien, biasanya dalam bentuk fitur yang lebih relevan.

c. Aplikasi
   1) Segmentasi Pelanggan: Membagi pelanggan menjadi kelompok berdasarkan perilaku pembelian mereka untuk pemasaran yang lebih efektif.
   2) Deteksi Anomali: Mendeteksi data yang berbeda secara signifikan dari data lainnya, seperti mendeteksi transaksi keuangan yang mencurigakan.

Unsupervised learning sangat berguna ketika kita ingin mengeksplorasi data, menemukan pola baru, atau ketika data berlabel tidak tersedia.

3. *Semi-Supervised Learning*

Semi-supervised learning adalah pendekatan yang menggabungkan elemen-elemen dari *supervised* dan *unsupervised learning*. Pendekatan ini digunakan ketika

sebagian kecil data pelatihan memiliki label, sedangkan sebagian besar lainnya tidak.

a. Proses: Model dilatih dengan menggunakan data berlabel dan tidak berlabel secara bersamaan. Data berlabel membantu memberikan bimbingan awal kepada model, sementara data tidak berlabel membantu model untuk mempelajari pola yang lebih umum dari data.
b. Contoh Algoritma
    1) *Label Propagation*: Teknik di mana label dari data berlabel disebarkan ke data yang tidak berlabel berdasarkan kemiripan fitur.
    2) *Co-Training*: Teknik di mana dua model dilatih pada subset fitur yang berbeda, dan kemudian hasil mereka digunakan untuk melabeli data tidak berlabel.
    3) *Self-Training*: Proses di mana model pertama kali dilatih pada data berlabel, kemudian digunakan untuk memprediksi data tidak berlabel, yang kemudian digunakan kembali untuk melatih model.
c. Aplikasi
    1) Pengenalan Wajah: Dalam dataset besar yang mungkin hanya memiliki beberapa gambar yang diberi label.
    2) Klasifikasi Dokumen: Ketika hanya sebagian kecil dokumen telah dikategorikan oleh manusia.

Semi-supervised learning sangat bermanfaat ketika pengumpulan data berlabel mahal atau memakan waktu, tetapi data tidak berlabel tersedia dalam jumlah besar.

4. *Reinforcement Learning*

Reinforcement Learning (RL) adalah kategori yang berbeda dari *supervised* dan *unsupervised learning*. Dalam pendekatan ini, agen belajar untuk mengambil tindakan dalam suatu lingkungan untuk memaksimalkan beberapa bentuk reward atau penghargaan.

a. Proses: Agen mengambil tindakan berdasarkan keadaan saat ini dan menerima umpan balik dari lingkungan dalam bentuk reward atau punishment. Tujuan agen adalah untuk belajar strategi (*policy*) yang memaksimalkan reward kumulatif dalam jangka panjang.
b. Contoh Algoritma
    1) *Q-Learning*: Algoritma yang belajar nilai dari tindakan dalam keadaan tertentu untuk mengembangkan kebijakan optimal.
    2) *Deep Q-Networks* (DQN): Menggabungkan Q-learning dengan deep learning untuk menangani situasi dengan ruang keadaan yang besar.
    3) *Policy Gradients*: Algoritma yang belajar kebijakan langsung, tanpa mengharuskan penilaian terhadap setiap tindakan secara eksplisit.
c. Aplikasi
    1) Permainan Video: Agen RL digunakan untuk mengalahkan pemain manusia dalam permainan seperti catur atau Go.
    2) Kendaraan Otonom: Digunakan untuk mengambil keputusan real-time dalam mengemudi.
    3) *Robotics*: Mengajar robot untuk berinteraksi dengan lingkungan fisik secara efektif.

Reinforcement learning sangat kuat dalam situasi di mana keputusan harus dibuat secara berurutan, dan hasilnya tidak langsung diketahui. Bagian ini memberikan penjelasan yang jelas tentang berbagai kategori *machine learning*, menjelaskan konsep, proses, contoh algoritma, dan aplikasi nyata dari setiap kategori. Dengan memahami perbedaan antara *supervised*, *unsupervised*, *semi-supervised*, dan *reinforcement learning*, pembaca dapat lebih tepat memilih pendekatan yang sesuai untuk berbagai jenis masalah yang dihadapi.

D. Proses Pengembangan Model *Machine Learning*

Pengembangan model *Machine Learning* (ML) adalah sebuah proses iteratif yang melibatkan beberapa tahap penting, mulai dari pengumpulan data hingga evaluasi dan penyempurnaan model. Setiap tahap memiliki peran yang krusial dalam memastikan bahwa model yang dibangun mampu memberikan prediksi atau keputusan yang akurat dan dapat diandalkan. Berikut ini adalah langkah-langkah utama dalam proses pengembangan model ML:

1. Pengumpulan Data

Tahap pertama dalam pengembangan model ML adalah pengumpulan data. Data adalah bahan bakar dari *machine learning*, dan kualitas data yang dikumpulkan akan sangat mempengaruhi kinerja model (mitchell, 1997).

a. Identifikasi Sumber Data: Langkah awal adalah menentukan sumber data yang relevan dengan masalah yang ingin diselesaikan. Data dapat berasal dari berbagai sumber, seperti basis data internal, API eksternal, file log, sensor IoT, atau bahkan data publik.

b. Mengumpulkan Data yang Cukup: Penting untuk mengumpulkan data dalam jumlah yang memadai, karena lebih banyak data biasanya membantu model untuk belajar lebih baik. Namun, jumlah data yang diperlukan dapat bervariasi tergantung pada kompleksitas masalah dan jenis algoritma yang digunakan.
c. Pertimbangan Etika Privasi: Saat mengumpulkan data, penting untuk memperhatikan aspek etika dan privasi, terutama jika data tersebut mengandung informasi pribadi. Kepatuhan terhadap regulasi seperti GDPR (*General Data Protection Regulation*) harus dijaga.

2. Pemrosesan dan Pembersihan Data

Setelah data dikumpulkan, langkah berikutnya adalah pemrosesan dan pembersihan data. Data mentah seringkali mengandung banyak noise, ketidakkonsistenan, dan ketidaksempurnaan yang harus diatasi sebelum data tersebut dapat digunakan untuk melatih model (domingos, 2015).

a. Pembersihan Data: Ini termasuk mengatasi data yang hilang (*missing values*), memperbaiki kesalahan dalam data, menghapus duplikat, dan menyelaraskan format data.
b. Normalisasi dan Standarisasi: Beberapa algoritma ML lebih sensitif terhadap skala data, sehingga penting untuk menormalkan atau menstandarisasi fitur numerik agar berada dalam rentang yang sama.
c. Transformasi Data: terkadang, data perlu ditransformasikan ke dalam bentuk yang lebih sesuai untuk analisis. Ini bisa termasuk encoding variabel kategori, ekstraksi fitur, atau penggunaan teknik seperti

Principal Component Analysis (PCA) untuk mengurangi dimensi data.

d. Data Augmentation: Untuk jenis data tertentu, seperti gambar atau teks, teknik augmentasi dapat digunakan untuk meningkatkan variasi data pelatihan tanpa harus mengumpulkan lebih banyak data baru. Misalnya, dalam pengenalan gambar, augmentasi bisa berupa rotasi, *flipping*, atau penambahan *noise*.

3. Pembagian Data

Agar model dapat dievaluasi secara obyektif, data biasanya dibagi menjadi dua (atau lebih) set yaitu *training set* dan *testing set* (shalev-shwartz and ben-david, 2014).

a. *Training Set*: Bagian data yang digunakan untuk melatih model. Model belajar dari pola dan hubungan dalam data ini.
b. *Testing Set*: Bagian data yang digunakan untuk menguji kinerja model setelah pelatihan. Data ini tidak digunakan selama pelatihan, sehingga memberikan gambaran yang lebih realistis tentang bagaimana model akan berkinerja pada data baru yang belum pernah dilihat sebelumnya.
c. *Validation Set*: Kadang-kadang, data juga dibagi menjadi validation set, yang digunakan untuk memilih dan menyetel model selama pelatihan, terutama ketika melakukan hyperparameter tuning.

Teknik *cross-validation* juga sering digunakan, di mana data dibagi menjadi beberapa subset dan model dilatih serta diuji pada kombinasi yang berbeda dari subset ini untuk mendapatkan evaluasi yang lebih andal.

4. Pemilihan dan Penggunaan Algoritma

Setelah data siap, langkah berikutnya adalah memilih algoritma *machine learning* yang tepat. Pemilihan algoritma bergantung pada beberapa faktor, termasuk jenis data, masalah yang akan diselesaikan, dan sumber daya komputasi yang tersedia (aggarwal, 2015).

a. *Supervised Learning*: Jika data memiliki label (*output* yang diketahui), algoritma seperti *regresi linear*, *decision tree*, atau *random forest* dapat digunakan. *Supervised learning* cocok untuk masalah seperti klasifikasi dan regresi.
b. *Unsupervised Learning*: Jika data tidak memiliki label, algoritma seperti *K-means clustering* atau PCA dapat digunakan untuk menemukan struktur atau pola dalam data. *Unsupervised learning* sering digunakan untuk *clustering*, pengelompokan, atau pengurangan dimensi.
c. *Semi-Supervised* dan *Reinforcement Learning*: Dalam beberapa kasus, baik *supervised* maupun *unsupervised learning* digunakan secara bersamaan (*semi-supervised*), atau algoritma *reinforcement learning* diterapkan untuk pembelajaran melalui percobaan dan *error*.
d. *Ensemble Methods*: Dalam beberapa situasi, menggabungkan beberapa model (misalnya, menggunakan *boosting*, *bagging*, atau *stacking*) dapat memberikan kinerja yang lebih baik daripada menggunakan satu model saja.

Setelah algoritma dipilih, model dilatih menggunakan data training. Proses ini melibatkan penyesuaian parameter model untuk meminimalkan kesalahan atau

memaksimalkan kinerja berdasarkan metrik yang telah ditentukan.

5. Evaluasi Model dan Fine-Tuning

Tahap terakhir dalam pengembangan model adalah evaluasi dan penyempurnaan model. Ini melibatkan pengujian model pada data *testing* dan, jika perlu, melakukan penyetelan (*fine-tuning*) untuk meningkatkan kinerjanya (marsland, 2014).

a. Evaluasi Kinerja: Model diuji pada data *testing*, dan metrik seperti akurasi, *precision*, *recall*, F1-*score*, MSE, atau AUC-ROC digunakan untuk mengukur kinerjanya. Jika kinerjanya tidak memuaskan, bisa jadi model mengalami *overfitting* atau *underfitting*.
b. *Hyperparameter Tuning*: Proses di mana parameter yang tidak dipelajari (*hyperparameter*) dari algoritma diatur untuk mengoptimalkan kinerja model. Teknik seperti *grid search* atau *random search* sering digunakan untuk mencari kombinasi *hyperparameter* terbaik.
c. *Regularization*: Jika model mengalami *overfitting*, teknik *regularization* seperti L1 (*Lasso*) atau L2 (*Ridge*) dapat diterapkan untuk mengurangi kompleksitas model dan meningkatkan kemampuan generalisasinya.
d. *Model Refinement*: Setelah evaluasi, model mungkin perlu disempurnakan dengan lebih banyak data, menggunakan algoritma yang berbeda, atau melalui perubahan dalam arsitektur atau *hyperparameter*.

Proses ini terus berlanjut secara iteratif sampai model mencapai kinerja yang memuaskan pada data testing. Model yang baik adalah model yang tidak hanya bekerja dengan baik pada data yang telah dilatih, tetapi

juga mampu menggeneralisasi dengan baik pada data baru. Bagian ini memberikan panduan komprehensif tentang langkah-langkah utama dalam mengembangkan model *machine learning*, menekankan pentingnya setiap tahap dalam membangun model yang efektif dan andal.

E. Tantangan dan Batasan *Machine Learning*

Meskipun *Machine Learning* (ML) telah memberikan dampak besar di berbagai bidang, teknologi ini juga dihadapkan pada sejumlah tantangan dan batasan yang perlu dipahami dan diatasi. Tantangan-tantangan ini berkaitan dengan data, algoritma, interpretabilitas, serta aspek etika dan privasi. Memahami tantangan dan batasan ini penting untuk mengembangkan dan menerapkan model ML yang efektif dan bertanggung jawab.

1. Kualitas dan Kuantitas Data

Data merupakan fondasi utama dari *machine learning*, dan kualitas serta kuantitas data yang digunakan sangat mempengaruhi kinerja model. Namun, pengelolaan data memiliki tantangan tersendiri (kelleher, mac namee and d'arcy, 2015).

a. Data yang Tidak Seimbang atau Tidak Memadai: Kurangnya data yang cukup atau data yang tidak seimbang (misalnya, lebih banyak contoh dari satu kelas dibandingkan kelas lain) dapat menyebabkan model belajar pola yang salah atau menjadi bias. Model yang dilatih dengan data tidak seimbang cenderung memberikan prediksi yang tidak akurat untuk kelas minoritas.

b. Data yang Kotor atau Tidak Lengkap: Data yang mengandung banyak noise, data yang hilang, atau anomali dapat menurunkan kinerja model. Pembersih-

an data (data cleaning) adalah langkah penting tetapi memakan waktu dan tidak selalu efektif dalam mengatasi semua masalah yang mungkin ada dalam data.

c. Data yang Tidak Relevan: Penggunaan data yang tidak relevan atau tidak terkait dengan masalah yang ingin diselesaikan dapat menurunkan kualitas model. Fitur-fitur yang tidak relevan dapat memperkenalkan noise yang mengganggu model dalam mengenali pola yang benar.

2. Kompleksitas Model dan Risiko *Overfitting*

Model ML yang kompleks dapat menangkap pola yang lebih halus dalam data, tetapi mereka juga rentan terhadap *overfitting* (chollet, 2018).

a. *Overfitting*: Terjadi ketika model terlalu rumit dan belajar terlalu banyak dari data pelatihan, termasuk *noise* atau pola yang kebetulan. Model yang *overfit* akan menunjukkan kinerja yang sangat baik pada data pelatihan tetapi buruk pada data baru yang belum pernah dilihat (data *testing*). *Overfitting* dapat diatasi dengan teknik regularisasi, *cross-validation*, atau dengan menggunakan model yang lebih sederhana.

b. *Underfitting*: Sebaliknya, *underfitting* terjadi ketika model terlalu sederhana untuk menangkap pola yang ada dalam data, menyebabkan kinerja yang buruk pada data pelatihan maupun data testing. Ini sering kali dapat diatasi dengan memilih model yang lebih kompleks atau dengan menambahkan lebih banyak fitur yang relevan.

c. *Dimensionality*: Model dengan terlalu banyak fitur (*dimensionality* tinggi) dapat menjadi terlalu kompleks

dan rentan terhadap *overfitting*, sementara model dengan terlalu sedikit fitur mungkin tidak cukup fleksibel untuk menangkap pola yang relevan. Teknik pengurangan dimensi seperti PCA (*Principal Component Analysis*) dapat digunakan untuk mengatasi masalah ini.

3. Interpretabilitas dan Transparansi Model

Salah satu tantangan utama dalam *machine learning*, terutama dengan model yang sangat kompleks seperti deep learning, adalah interpretabilitas (dua and graff, 2019).

a. *Black-Box Models*: Banyak model ML, terutama *deep learning*, dikenal sebagai "*black-box models*" karena sulit dipahami bagaimana model mencapai suatu keputusan atau prediksi. Kurangnya transparansi ini menjadi masalah besar di bidang-bidang yang membutuhkan kepercayaan dan penjelasan, seperti dalam aplikasi medis, keuangan, dan hukum.

b. *Explainable* AI (XAI)**:** Ada dorongan kuat dalam komunitas ML untuk mengembangkan teknik *Explainable* AI, yang bertujuan membuat model ML lebih dapat dipahami oleh manusia. Teknik ini termasuk LIME (*Local Interpretable Model-agnostic Explanations*) dan SHAP (*SHapley Additive exPlanations*), yang memberikan wawasan tentang bagaimana model membuat keputusan.

c. Bias dalam Model**:** Model ML dapat mewarisi atau memperburuk bias yang ada dalam data pelatihan, yang dapat menyebabkan keputusan yang tidak adil atau diskriminatif. Identifikasi dan mitigasi bias adalah

tantangan besar yang membutuhkan perhatian ekstra dalam pengembangan model.

4. Tantangan Etika dan Privasi

Penggunaan data dalam *machine learning* menimbulkan sejumlah tantangan etika dan privasi yang perlu diperhatikan (vapnik, 1998).

a. *Data Privacy*: Dalam banyak kasus, data yang digunakan untuk melatih model ML mencakup informasi pribadi yang sensitif. Melindungi privasi individu adalah tantangan besar, terutama dalam era di mana data besar (*big data*) sangat berharga. Regulasi seperti GDPR (*General Data Protection Regulation*) di Eropa memberikan pedoman ketat tentang bagaimana data pribadi harus dikelola dan digunakan.
b. Etika dalam Penggunaan Model ML: Penggunaan model ML dalam pengambilan keputusan otomatis (*automated decision-making*) menimbulkan pertanyaan etika, terutama ketika model tersebut dapat mempengaruhi kehidupan orang secara signifikan, seperti dalam rekrutmen, pemberian pinjaman, atau penegakan hukum. Terdapat risiko bahwa model ML dapat digunakan secara tidak adil atau tidak transparan, yang menuntut pengawasan dan pengendalian yang ketat.
c. *Fairness and Accountability*: Terdapat kebutuhan untuk memastikan bahwa model ML adil dan tidak diskriminatif. Ini mencakup perlunya pengujian model untuk bias dan ketidakadilan, serta pengembangan kebijakan yang mendorong akuntabilitas dalam penerapan model ML.

5. Keterbatasan Sumber Daya dan Skalabilitas

Pengembangan dan penerapan model ML yang efektif sering kali membutuhkan sumber daya yang signifikan (zhang, 2017).

a. Kebutuhan Komputasi yang Tinggi: Banyak algoritma ML, terutama deep learning, membutuhkan daya komputasi yang sangat besar untuk pelatihan dan inferensi. Ini bisa menjadi penghalang, terutama bagi organisasi kecil atau individu yang tidak memiliki akses ke perangkat keras yang canggih seperti GPU atau TPU.
b. Skalabilitas Model: Saat data yang digunakan bertambah besar, model ML harus dapat diskalakan agar tetap efektif. Skalabilitas mencakup tidak hanya kemampuan model untuk menangani volume data yang besar, tetapi juga kemampuannya untuk melakukan prediksi dalam waktu yang wajar, terutama dalam aplikasi *real-time*.
c. Pengelolaan dan Pemeliharaan Model**:** Model ML yang sudah dilatih sering kali memerlukan pemeliharaan berkelanjutan, termasuk pemantauan kinerja, pembaruan data pelatihan, dan penyetelan ulang model seiring perubahan dalam data atau konteks aplikasi. Ini bisa menjadi tantangan besar dalam lingkungan produksi.

Bagian ini memberikan gambaran tentang berbagai tantangan dan batasan yang dihadapi dalam pengembangan dan penerapan *machine learning*. Dengan memahami dan mengatasi tantangan-tantangan ini, praktisi ML dapat mengembangkan model yang lebih

kuat, adil, dan dapat dipercaya, serta menerapkannya secara bertanggung jawab.

F. Studi Kasus: Penerapan *Machine Learning* di Industri

Machine Learning (ML) telah menjadi kekuatan pendorong di berbagai industri, membantu perusahaan untuk membuat keputusan yang lebih baik, meningkatkan efisiensi, dan menciptakan produk serta layanan baru. Dengan kemampuannya untuk menganalisis data besar dan menemukan pola yang kompleks, ML menawarkan solusi inovatif untuk tantangan bisnis dan operasional. Berikut ini adalah beberapa penerapan utama ML di berbagai sektor industri:

1. Industri Keuangan

Industri keuangan adalah salah satu pengguna terbesar *Machine Learning*. ML digunakan untuk meng-otomatisasi proses, meningkatkan layanan pelanggan, dan mendeteksi aktivitas yang mencurigakan.

a. Deteksi Penipuan: ML digunakan untuk menganalisis transaksi keuangan secara real-time untuk mendeteksi pola yang mencurigakan atau anomali yang mungkin menunjukkan penipuan. Algoritma seperti jaringan saraf dan random forest sering digunakan untuk memodelkan dan mendeteksi transaksi yang tidak biasa.

b. Penilaian Kredit: Perusahaan keuangan menggunakan ML untuk mengevaluasi risiko kredit calon peminjam. Dengan menganalisis data historis dan pola perilaku, model ML dapat memberikan penilaian yang lebih akurat daripada metode tradisional.

c. Manajemen Portofolio: ML juga diterapkan dalam manajemen portofolio untuk mengoptimalkan alokasi

aset dan memprediksi pergerakan pasar. Algoritma pembelajaran mendalam (*deep learning*) digunakan untuk menganalisis data pasar yang kompleks dan menemukan peluang investasi.

d. Pengalaman Pelanggan: Chatbot dan asisten virtual berbasis ML digunakan oleh bank dan perusahaan keuangan untuk menyediakan layanan pelanggan 24/7, menjawab pertanyaan, dan membantu dalam berbagai transaksi.

2. Industri Kesehatan

Dalam industri kesehatan, *machine learning* memiliki potensi untuk meningkatkan hasil klinis, mempercepat penelitian, dan mengurangi biaya.

a. Diagnosis Penyakit**:** ML digunakan untuk menganalisis data medis, seperti gambar radiologi atau rekam medis elektronik, untuk membantu dokter dalam mendiagnosis penyakit. Model deep learning, seperti *convolutional neural networks* (CNN), telah menunjukkan keunggulan dalam mengidentifikasi pola pada gambar medis yang sulit dikenali oleh manusia.

b. Pengembangan Obat: Proses pengembangan obat dapat dipercepat dengan ML, yang digunakan untuk memprediksi efek obat, menemukan target molekuler baru, dan memodelkan interaksi kimia. Dengan memanfaatkan data dari penelitian sebelumnya, ML membantu mengurangi waktu dan biaya yang diperlukan untuk membawa obat baru ke pasar.

c. Personalisasi Pengobatan: ML digunakan untuk mengembangkan pendekatan pengobatan yang disesuaikan dengan kebutuhan individu pasien. Ini

mencakup analisis genomik untuk menentukan terapi yang paling efektif berdasarkan profil genetik pasien.

d. Manajemen Rumah Sakit: ML juga diterapkan untuk mengoptimalkan operasi rumah sakit, seperti manajemen sumber daya, penjadwalan pasien, dan prediksi permintaan layanan kesehatan.

3. Industri Ritel

Industri ritel menggunakan *machine learning* untuk meningkatkan pengalaman pelanggan, mengoptimalkan inventaris, dan meningkatkan efisiensi operasional.

a. Rekomendasi Produk: Sistem rekomendasi berbasis ML digunakan oleh pengecer online untuk menyarankan produk kepada pelanggan berdasarkan riwayat pembelian mereka dan perilaku penelusuran. Algoritma seperti *collaborative filtering* dan *content-based filtering* digunakan untuk mempersonalisasi pengalaman belanja.

b. Manajemen Inventaris: ML membantu pengecer mengoptimalkan manajemen inventaris dengan memprediksi permintaan produk, mengidentifikasi tren musiman, dan mengurangi stok berlebih. Ini memungkinkan pengecer untuk mengurangi biaya penyimpanan dan meningkatkan ketersediaan produk.

c. Pemasaran yang Dipersonalisasi: Pengecer menggunakan ML untuk mempersonalisasi kampanye pemasaran, mengirimkan penawaran yang disesuaikan kepada pelanggan berdasarkan preferensi dan perilaku mereka. Analisis prediktif digunakan untuk menentukan kapan dan bagaimana berinteraksi dengan pelanggan untuk memaksimalkan konversi.

d. Pengoptimalan Harga: ML digunakan untuk menyesuaikan harga produk secara dinamis berdasarkan permintaan pasar, persaingan, dan perilaku pelanggan. Ini membantu pengecer tetap kompetitif sambil memaksimalkan margin keuntungan.

4. Industri Transportasi dan Logistik

Industri transportasi dan logistik telah melihat peningkatan efisiensi yang signifikan berkat penerapan *Machine Learning*.

a. Pemeliharaan Prediktif: ML digunakan untuk memprediksi kegagalan peralatan pada kendaraan atau infrastruktur transportasi, memungkinkan pemeliharaan dilakukan sebelum masalah terjadi. Ini mengurangi *downtime* dan biaya perbaikan.
b. Optimasi Rute: Perusahaan logistik menggunakan ML untuk mengoptimalkan rute pengiriman, mempertimbangkan faktor-faktor seperti kondisi lalu lintas, cuaca, dan pola permintaan. Ini membantu mengurangi waktu pengiriman dan biaya bahan bakar.
c. Kendaraan Otonom: ML adalah komponen inti dari teknologi kendaraan otonom, di mana model pembelajaran mendalam digunakan untuk memahami lingkungan sekitar kendaraan, membuat keputusan mengemudi, dan meningkatkan keselamatan.
d. Manajemen Rantai Pasokan: ML diterapkan untuk mengelola rantai pasokan, termasuk peramalan permintaan, pengelolaan stok, dan pengoptimalan logistik. Ini memungkinkan perusahaan untuk merespons perubahan permintaan dengan cepat dan efisien.

5. Industri Manufaktur

Dalam industri manufaktur, *Machine Learning* digunakan untuk meningkatkan efisiensi produksi, mengurangi biaya, dan meningkatkan kualitas produk.

a. Pemeliharaan Prediktif: Seperti dalam transportasi, ML digunakan untuk memprediksi kapan mesin atau peralatan mungkin gagal, memungkinkan intervensi tepat waktu untuk mencegah downtime. Algoritma ML menganalisis data sensor dari mesin untuk mendeteksi pola kegagalan potensial.
b. Otomatisasi Kualitas: ML digunakan untuk memantau dan memastikan kualitas produk dalam proses produksi. Algoritma pengenalan gambar dapat mendeteksi cacat produk secara real-time, memungkinkan perbaikan segera.
c. Pengoptimalan Proses: ML membantu dalam meng-optimalkan proses produksi dengan menganalisis data dari seluruh lini produksi untuk menemukan cara meningkatkan efisiensi, mengurangi limbah, dan meningkatkan throughput.
d. Desain Produk: ML juga digunakan dalam fase desain produk, di mana model prediktif membantu insinyur merancang produk yang lebih baik dengan mem-prediksi kinerja dan kegagalan berdasarkan data historis.

Bagian ini menunjukkan bagaimana *Machine Learning* telah diterapkan secara luas di berbagai industri, masing-masing dengan tantangan uniknya sendiri tetapi dengan potensi besar untuk mengubah cara perusahaan beroperasi. Penerapan ML di sektor-sektor ini menunjuk-kan fleksibilitas dan kekuatan teknologi ini dalam men-

ciptakan nilai yang nyata dan terukur bagi bisnis di seluruh dunia.

Daftar Pustaka

Aggarwal, C. 2015. *Data Mining: The Textbook*. Jerman: Springer.

Bishop, C. 2006. *Pattern Recognition and Machine Learning*. Jerman: Springer.

Chollet, F. 2018. *Deep Learning with Python*. Amerika Serikat: Manning Publications.

Domingos, P. 2015. *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Amerika Serikat: Basic Books.

Dua, D. and Graff, C. 2019. *UCI Machine Learning Repository*. Amerika Serikat: University of California, Irvine, School of Information and Computer Sciences.

Goodfellow, I., Bengio, Y. and Courville, A. 2016. *Deep Learning*. Amerika Serikat: MIT Press.

Hastie, T., Tibshirani, R. and Friedman, J. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Jerman: Springer.

Kelleher, J.., Mac Namee, B. and D'Arcy, A. 2015. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. Amerika Serikat: MIT Press.

Marsland, S. 2014. *Machine Learning_An Algorithmic Perspective_1stEdition*. Amerika Serikat: CRC Press.

Mitchell, T. 1997. *Machine Learning*. Amerika Serikat: McGraw-Hill.

Murphy, K. 2012. *Machine Learning: A Probabilistic Perspective*. Amerika Serikat: MIT Press.

Russell, S. and Norvig, P. (2020) *Artificial Intelligence: A Modern Approach*. Inggris: Pearson.

Shalev-Shwartz, S. and Ben-David, S. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Inggris: Cambridge University Press.

Vapnik, V. 1998. *Statistical Learning Theory*. Amerika Serikat: Wiley.

Zhang, Z. 2017. *A Gentle Introduction to Machine Learning*. Amerika Serikat: Independent Publisher.

BAB 2

Supervised Learning

-Ratnasari-

A. Pengertian Supervised Learning

Supervised learning adalah salah satu metode dalam pembelajaran mesin (**machine learning**) di mana model dilatih menggunakan data yang diberi label. Dalam pendekatan ini, setiap contoh dalam data pelatihan memiliki input dan output yang diharapkan (label). Tujuan dari supervised learning adalah memprediksi output atau label berdasarkan input yang baru setelah mempelajari pola dari data yang telah diberi label.

Menurut Goodfellow *et al.* (2016), "*Supervised learning involves learning a function that maps an input to an output based on example input-output pairs*." Hal ini memungkinkan model untuk mempelajari hubungan antara input dan output sehingga dapat memprediksi dengan akurat untuk data baru.

Berikut perbedaan antara supervised learning, unsupervised learning, dan reinforcement learning:

1. **Supervised Learning**

   Cara kerja supervised Learning adalah dengan melatih model menggunakan data yang diberi label, artinya setiap contoh dalam data pelatihan memiliki input dan output (label) yang diketahui.

   Tujuan dari Supervised Learning adalah memprediksi label atau output berdasarkan input baru setelah model belajar dari pola data yang ada.

   Tipe Masalah:

   Klasifikasi: Memisahkan data ke dalam beberapa kelas (contoh: mendeteksi email spam).

   Regresi: Memprediksi nilai kontinu (contoh: memprediksi harga rumah).

   Contoh Algoritma yang digunakan Decision Tree, Support Vector Machine (SVM), Linear Regression.

2. **Unsupervised Learning**

   Unsupervised Learning adalah salah satu jenis pembelajaran mesin (machine learning) di mana model dilatih menggunakan data yang tidak diberi label. Artinya, model tidak memiliki output yang diketahui dan bertujuan menemukan pola atau struktur tersembunyi dari data. Tujuan dari Unsupervised Learning adalah menemukan pola atau struktur data seperti kluster, asosiasi, atau pengurangan dimensi.

   Tipe Masalah:

   Clustering: Mengelompokkan data ke dalam kelompok yang mirip (contoh: segmentasi pasar).

   Association: Menemukan hubungan antar variabel dalam data (contoh: market basket analysis).

Contoh Algoritma yang digunakan K-Means Clustering, Hierarchical Clustering, Principal Component Analysis (PCA).

Contoh Kasus: Segmentasi pelanggan, rekomendasi produk, pengelompokan dokumen.

3. **Reinforcement Learning**

Reinforcement Learning adalah salah satu jenis pembelajaran mesin di mana model belajar melalui interaksi dengan lingkungan. Model mendapatkan umpan balik berupa reward atau punishment berdasarkan tindakan yang diambil, dengan tujuan memaksimalkan total reward dalam jangka panjang.

Tujuan dari Reinforcement Learning adalah mempelajari kebijakan (policy) optimal yang memandu pengambilan keputusan di berbagai keadaan untuk memaksimalkan reward.

Proses: Agent mengambil tindakan di lingkungan, menerima feedback, dan memperbaiki tindakan di masa depan berdasarkan feedback tersebut.

Contoh Algoritma: Q-Learning, Deep Q-Network (DQN), Policy Gradient.

Contoh Kasus: Permainan video (contoh: AlphaGo), robotika, sistem rekomendasi dinamis.

B. Jenis-jenis Masalah dalam Supervised Learning

Dalam supervised learning, terdapat dua jenis utama masalah yang dihadapi berdasarkan tipe output yang diprediksi oleh model. Berikut adalah penjelasan tentang kedua jenis masalah tersebut:

1. **Efektifitas Kalsifikasi (Clasification)**

Klasifikasi adalah salah satu teknik dalam pembelajaran mesin (*machine learning*) yang bertujuan untuk mengkategorikan atau mengklasifikasikan data ke dalam label atau kelas yang telah ditentukan sebelumnya. Algoritma klasifikasi bekerja dengan mempelajari pola dari data berlabel selama tahap pelatihan, kemudian menggunakan pola tersebut untuk mengklasifikasikan data baru yang belum dilabeli (James *et al.,* 2013).

Proses klasifikasi biasanya melibatkan dua tahap utama yaitu:

a. Training (Pelatihan): Model dilatih menggunakan data yang sudah diberi label, sehingga model dapat mengenali pola dalam data.
b. Prediction (Prediksi): Setelah pelatihan selesai, model digunakan untuk memprediksi kelas dari data baru yang belum diberi label.

Contoh algoritma klasifikasi adalah:

K-Nearest Neighbors (KNN): Mengklasifikasikan objek berdasarkan seberapa dekat objek tersebut dengan objek lain dalam data latih.

Decision Tree: Menggunakan struktur seperti pohon untuk mengambil keputusan berdasarkan fitur-fitur dalam data.

Support Vector Machine (SVM): Menemukan hyperplane yang memisahkan data dari berbagai kelas dengan margin terbesar.

Naive Bayes: Berdasarkan teorema Bayes, mengasumsikan bahwa setiap fitur dalam data berkontribusi secara independen terhadap probabilitas kelas.

Logistic Regression: Digunakan untuk klasifikasi biner, memodelkan probabilitas suatu kelas berdasarkan fungsi logistik.

Random Forest: Algoritma ini merupakan pengembangan dari Decision Tree, di mana ia membangun banyak pohon keputusan (hutan) dan menggabungkan hasil dari setiap pohon untuk membuat prediksi akhir. Random Forest sangat efektif dalam mengurangi overfitting dan meningkatkan akurasi model karena menggabungkan prediksi dari banyak pohon keputusan.

Contoh Kasus dalam Klasifikasi:

Deteksi Penyakit: Memprediksi apakah pasien memiliki diabetes (kelas "yes" atau "no") berdasarkan faktor kesehatan.

Pengenalan Gambar: Klasifikasi gambar sebagai "kucing", "anjing", atau "burung".

Analisis Sentimen: Menentukan apakah ulasan produk bersifat "positif", "negatif", atau "netral".

2. Regresi (*Regression*)

Regresi (*regression*) adalah metode dalam pembelajaran mesin yang digunakan untuk memodelkan hubungan antara variabel independen (fitur) dan variabel dependen (target) dengan tujuan memprediksi nilai kontinu. Berbeda dengan klasifikasi, yang memprediksi kelas atau kategori, regresi digunakan ketika target yang diprediksi berupa nilai numerik atau kuantitatif (Hastie *et al*., 2009).

Beberapa jenis Algoritma regresi yang umum digunakan:

Linear Regression: Regresi ini memodelkan hubungan linear antara variabel independen dan dependen

dengan mencoba menemukan garis lurus terbaik yang meminimalkan kesalahan antara nilai yang diprediksi dan nilai aktual.

Contoh: Memprediksi harga rumah berdasarkan ukuran dan lokasi rumah.

Polynomial Regression: Merupakan perpanjangan dari regresi linear yang memungkinkan hubungan non-linear antara variabel independen dan dependen dengan memasukkan polinomial pada fitur.

Contoh: Memprediksi kinerja mesin berdasarkan suhu dan tekanan yang mungkin memiliki hubungan non-linear.

Ridge Regression: Varian dari linear regression yang menambahkan penalti pada besarnya koefisien untuk mengatasi overfitting.

Lasso Regression: Seperti Ridge Regression, tetapi Lasso juga melakukan seleksi fitur dengan menekan koefisien fitur yang tidak signifikan menjadi nol.

Logistic Regression: Meskipun sering digunakan untuk klasifikasi biner, logistic regression secara teknis adalah model regresi karena memprediksi probabilitas suatu peristiwa terjadi, dan hasilnya digunakan untuk menentukan kelas.

Support Vector Regression (SVR): Algoritma Support Vector Machine yang diadaptasi untuk tugas regresi, yang berusaha memprediksi nilai kontinu dengan mempertahankan margin antara prediksi dan nilai sebenarnya.

Regresi sering digunakan dalam aplikasi seperti memprediksi harga saham, pertumbuhan ekonomi, biaya medis, dan lain-lain.

C. Algoritma Supervised Learning

Berikut algoritma suvervised learning yang sering digunakan:

1. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi, dengan prinsip dasar mengklasifikasikan data berdasarkan kedekatannya dengan data lain yang sudah diberi label (Kotsiantis, 2006).

Prinsip Kerja K-Nearest Neighbors (KNN) adalah:

a. Menentukan Nilai K: Pilih jumlah tetangga terdekat (K) yang akan dipertimbangkan dalam proses klasifikasi atau regresi.
b. Mengukur Jarak: Hitung jarak antara data baru dan semua data dalam dataset pelatihan, biasanya menggunakan jarak Euclidean.
c. Menentukan Tetangga Terdekat: Pilih K data terdekat berdasarkan jarak yang dihitung.

Kelebihan KNN

a. Sederhana dan Mudah Dipahami: KNN mudah diterapkan dan tidak memerlukan model yang kompleks.
b. Non-parametrik: Tidak memerlukan pemodelan eksplisit, memberikan fleksibilitas dalam menangani data.
c. Kinerja Baik pada Data Tidak Linear: Dapat bekerja efektif dengan data non-linear.

Kekurangan KNN

a. Kinerja pada Data Besar: Kurang efisien pada dataset besar karena perhitungan jarak yang memakan waktu.
b. Sensitif terhadap Fitur Tidak Relevan: Terpengaruh oleh fitur yang tidak relevan jika tidak dilakukan pemilihan fitur yang baik.
c. Memerlukan Penyimpanan Data: Menyimpan seluruh dataset pelatihan untuk proses klasifikasi atau prediksi.
d. Pemilihan Nilai K yang Optimal: Menentukan nilai K yang optimal sering memerlukan validasi silang.

Kasus Penggunaan

a. Klasifikasi Dokumen: Mengklasifikasikan dokumen ke dalam kategori seperti spam atau tidak spam.
b. Pengenalan Wajah: Identifikasi wajah berdasarkan kemiripan dengan wajah dalam database.
c. Rekomendasi Produk: Memberikan rekomendasi berdasarkan kesamaan preferensi.
d. Diagnosis Medis: Klasifikasi penyakit berdasarkan gejala mirip data pasien lain.

2. Decision Tree

Decision tree adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. Model ini membentuk struktur pohon di mana setiap simpul mewakili keputusan berdasarkan fitur data, dan setiap cabang mewakili hasil keputusan tersebut (Quinlan, J.R. (1986).

Struktur Decision Tree:

a. Akar (*Root*): Simpul pertama dari pohon yang mewakili seluruh dataset. Akar memutuskan fitur mana yang akan digunakan untuk pembagian awal.
b. Simpul Internal (*Internal Nodes*): Simpul yang mewakili keputusan berdasarkan nilai fitur tertentu. Setiap simpul internal memiliki cabang yang mewakili hasil dari keputusan tersebut.
c. Simpul Daun (*Leaf Nodes*): Simpul terminal yang mewakili prediksi akhir atau kelas dari data yang telah dibagi.

Cara Kerja:

a. **Pembagian Data**: Pada setiap simpul, data dibagi berdasarkan fitur yang memberikan pemisahan terbaik, menggunakan kriteria tertentu untuk memilih fitur dan titik pemisahan yang optimal.
b. **Kriteria Pembagian**:
    1) Entropy dan Gain Informasi: Digunakan dalam algoritma ID3 dan C4.5 untuk mengukur seberapa banyak informasi yang diperoleh dari pembagian data berdasarkan fitur.
    2) Gini Impurity: Digunakan dalam algoritma CART, mengukur ketidakpastian kelas di simpul dan memilih fitur yang meminimalkan impurity.
    3) Mean Squared Error: Digunakan dalam regresi untuk memilih fitur yang meminimalkan rata-rata kuadrat kesalahan prediksi.
c. **Pemangkasan (Pruning)**: Setelah pohon dibangun, pemangkasan dilakukan untuk mengurangi kompleksitas model dan mencegah *overfitting* dengan menghilangkan simpul yang tidak signifikan.

Kelebihan Decision Tree:

a. Mudah Dipahami dan Diinterpretasikan: Struktur pohon memudahkan pemahaman dan interpretasi keputusan yang dibuat oleh model.
b. Tidak Memerlukan Prabaca Data: Tidak memerlukan normalisasi atau skala fitur karena keputusan dibuat berdasarkan nilai fitur.
c. Dapat Menangani Data Kategorikal dan Numerik: Mampu bekerja dengan berbagai jenis data tanpa transformasi tambahan.

Kekurangan:

a. Overfitting: Decision Tree yang terlalu dalam dapat menyesuaikan data pelatihan dengan sangat baik tetapi kurang dalam generalisasi pada data baru.
b. Instabilitas: Perubahan kecil dalam data pelatihan dapat menyebabkan perubahan besar dalam struktur pohon.
c. Bias Terhadap Fitur dengan Banyak Nilai: Fitur dengan banyak nilai unik cenderung mendominasi pembagian karena lebih banyak kemungkinan pemisahan.

3. Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. SVM berusaha menemukan hyperplane terbaik yang memisahkan data ke dalam kelas-kelas yang berbeda dengan margin terbesar (Schölkopf, B., & Smola, A.J. (2002).

Cara Kerja SVM adalah sebagai berikut:

a. Hyperplane: Dalam ruang fitur, SVM mencari hyperplane yang memisahkan data dari dua kelas dengan margin maksimum. Hyperplane ini adalah garis atau permukaan yang memisahkan kelas-kelas dengan jarak maksimum dari titik-titik data terdekat dari kedua kelas, yang disebut support vectors.
b. Margin: Margin adalah jarak antara hyperplane dan titik-titik data terdekat dari kelas-kelas yang berbeda. Tujuan SVM adalah memaksimalkan margin ini untuk meningkatkan kemampuan generalisasi model.
c. SVM Linear: Untuk data yang dapat dipisahkan secara linear, SVM menemukan hyperplane terbaik yang memisahkan kelas-kelas dengan jarak terbesar. Dalam kasus ini, data sudah berada dalam ruang dimensi yang cukup untuk dipisahkan oleh hyperplane.
d. SVM Non-Linear: Untuk data yang tidak dapat dipisahkan secara linear, SVM menggunakan teknik kernel untuk memetakan data ke ruang fitur yang lebih tinggi di mana data menjadi linier terpisah. Ini dilakukan dengan fungsi kernel yang mengubah data ke dimensi yang lebih tinggi.

Penerapan Kernel

SVM menggunakan fungsi kernel untuk menangani data yang tidak dapat dipisahkan secara linear dengan cara berikut:

a. Kernel Linear: Menggunakan produk dot standar antara vektor fitur. Ini adalah kasus dasar ketika data sudah linier terpisah.
b. Polynomial Kernel: Menggunakan fungsi polinomial untuk memetakan data ke ruang fitur yang lebih

tinggi. Kernel ini dapat menangani hubungan non-linear antara fitur dan target.

c. Radial Basis Function (RBF) Kernel: Juga dikenal sebagai Gaussian kernel, mengukur jarak antara titik data dalam ruang fitur dan menerapkan fungsi Gaussian untuk membuat data lebih mudah dipisahkan. Ini adalah kernel yang sangat fleksibel dan sering digunakan untuk data yang sangat non-linear.
d. Sigmoid Kernel: Menggunakan fungsi sigmoid yang mirip dengan fungsi aktivasi dalam jaringan saraf, memungkinkan pemisahan data dengan fungsi non-linear.

4. Neural Networks

Neural Networks atau Jaringan syaraf tiruan adalah algoritma pembelajaran mesin yang terinspirasi oleh struktur dan fungsi otak manusia. Neural Networks digunakan untuk berbagai tugas pembelajaran mesin seperti klasifikasi, regresi, dan pengenalan pola. Neural Networks merupakan model komputasi yang terdiri dari sejumlah unit pemrosesan yang disebut neuron, yang diorganisasikan dalam lapisan-lapisan. Model ini dirancang untuk mempelajari representasi data dan membuat prediksi atau keputusan berdasarkan data tersebut(Bishop, C.M. 2006).

Prinsip Kerja Neural Networks sebagai berikut:

a. Input: Data dimasukkan ke dalam jaringan melalui lapisan input. Setiap neuron dalam lapisan input mewakili fitur dari data.
b. Proses:
    1) Lapisan Tersembunyi: Data diproses melalui satu atau lebih lapisan tersembunyi. Setiap neuron di

lapisan tersembunyi menghitung output berdasarkan fungsi aktivasi dari input yang diterima dan bobot yang diterapkan.

2) Fungsi Aktivasi: Fungsi yang diterapkan pada output dari neuron untuk menentukan apakah neuron akan diaktifkan atau tidak. Fungsi aktivasi yang umum digunakan termasuk sigmoid, ReLU (Rectified Linear Unit), dan tanh (hyperbolic tangent).

c. Output: Hasil akhir dari jaringan adalah output yang dihasilkan oleh lapisan output. Output ini dapat berupa kelas untuk masalah klasifikasi atau nilai kontinu untuk masalah regresi.

d. Pelatihan:

Forward Propagation: Data diproses dari lapisan input melalui lapisan tersembunyi hingga mencapai lapisan output untuk menghasilkan prediksi.

Backward Propagation: Proses untuk memperbarui bobot jaringan dengan menghitung gradien dari fungsi kehilangan (loss function) dan menerapkan algoritma optimisasi seperti Gradient Descent untuk mengurangi kesalahan prediksi.

Struktur Lapisan dan Bobot

a. Lapisan Input

Mengandung neuron yang sesuai dengan fitur dari data input.

b. Lapisan Tersembunyi

Beberapa lapisan ini mungkin ada dalam jaringan, dan setiap lapisan terdiri dari sejumlah neuron. Neuron-neuron ini mengolah data dengan menerapkan bobot dan fungsi aktivasi.

c. Lapisan Output

Mengandung neuron yang menghasilkan prediksi akhir. Jumlah neuron dalam lapisan output bergantung pada jenis masalah, misalnya, satu neuron untuk regresi atau sejumlah neuron yang sesuai dengan jumlah kelas untuk klasifikasi.

d. Bobot

Bobot adalah parameter yang menghubungkan neuron di satu lapisan dengan neuron di lapisan berikutnya. Bobot ini diperbarui selama pelatihan untuk meminimalkan kesalahan prediksi. Bobot awalnya diinisialisasi secara acak dan diperbarui selama proses pelatihan menggunakan teknik seperti Gradient Descent.

5. Ensemble Methods

Adalah teknik dalam pembelajaran mesin yang menggabungkan beberapa model untuk meningkatkan kinerja prediksi dibandingkan dengan model tunggal. Tujuan utama metode ensemble adalah untuk mengurangi kesalahan dan meningkatkan akurasi dengan memanfaatkan kekuatan dari beberapa model. Dua pendekatan utama dalam metode ensemble adalah Bagging dan Boosting (Hastie, T., Tibshirani, R., & Friedman, J. (2009).

Bagging

Bagging (*bootstrap aggregating*) adalah metode ensemble yang mengurangi variabilitas model dengan menggabungkan beberapa model yang dilatih pada subset acak dari data pelatihan (Breiman, L. (1996). Teknik ini bekerja dengan prinsip berikut:

a. Pembuatan Subset: Mengambil beberapa subset acak dari data pelatihan dengan pengembalian (bootstrap sampling), yaitu setiap subset mungkin mengandung beberapa pengulangan data.
b. Pelatihan Model: Melatih model yang sama pada setiap subset data. Model-model ini bisa berupa model sederhana seperti pohon keputusan.
c. Agregasi: Menggabungkan prediksi dari semua model untuk mendapatkan hasil akhir. Untuk klasifikasi, biasanya digunakan suara mayoritas (majority voting), dan untuk regresi, digunakan rata-rata dari prediksi model.

Contoh: Random Forest

Random Forest adalah salah satu algoritma yang menggunakan teknik bagging dengan pohon keputusan sebagai model dasar. Dalam Random Forest:

a. Pengacakan Fitur: Selama pelatihan, setiap pohon keputusan dalam hutan dibangun menggunakan subset acak dari fitur, tidak hanya data.
b. Agregasi: Prediksi akhir diperoleh dengan menggabungkan hasil dari semua pohon dalam hutan, baik melalui suara mayoritas untuk klasifikasi atau rata-rata untuk regresi.

Boosting

Boosting adalah metode ensemble yang membangun model secara berturut-turut, di mana setiap model baru berusaha memperbaiki kesalahan dari model sebelumnya. Prinsip dasar boosting meliputi:

a. Pelatihan Berturut-turut: Model dibangun satu per satu. Setiap model baru memberi bobot lebih pada data yang salah klasifikasi oleh model sebelumnya.
b. Kombinasi Model: Prediksi akhir diperoleh dengan menggabungkan prediksi dari semua model secara bertahap. Bobot dapat diberikan pada model berdasarkan kinerjanya.

Contoh: AdaBoost dan Gradient Boosting

AdaBoost (Adaptive Boosting): Menambahkan model secara iteratif dan memberikan bobot lebih pada data yang salah klasifikasi oleh model sebelumnya. Model akhir adalah gabungan dari model-model yang ada dengan bobot yang ditentukan oleh kinerjanya.

Gradient Boosting: Membangun model secara berturut-turut dengan mengurangi kesalahan residu dari model sebelumnya. Setiap model baru dilatih untuk memperbaiki kesalahan model sebelumnya, dan model akhir diperoleh dengan menjumlahkan hasil dari semua model.

D. Evaluasi Model

Evaluasi Model adalah proses untuk menilai kinerja model pembelajaran mesin setelah pelatihan untuk memastikan bahwa model tersebut bekerja dengan baik pada data yang belum pernah dilihat sebelumnya. Evaluasi model membantu menentukan sejauh mana model dapat melakukan generalisasi dan memberikan informasi tentang kekuatan serta kelemahan model (Friedman, J., Hastie, T., & Tibshirani, R. (2001).

Tujuan Evaluasi Model

1. Menilai Kinerja: Menentukan seberapa baik model melakukan tugas yang diinginkan, seperti klasifikasi atau regresi.
2. Memilih Model Terbaik: Bandingkan berbagai model atau konfigurasi model untuk memilih yang paling sesuai untuk aplikasi tertentu.
3. Mengidentifikasi Masalah: Mengidentifikasi area di mana model mungkin kurang performa, seperti bias, varian, atau kesalahan sistematis.
4. Mengukur Kemampuan Generalisasi: Menilai seberapa baik model dapat menangani data yang tidak terlihat selama pelatihan.

Metode Evaluasi Model

1. Cross-Validation: Proses membagi data menjadi beberapa subset atau "fold", di mana model dilatih pada subset data tertentu dan diuji pada subset data yang berbeda. Teknik ini memberikan gambaran yang lebih stabil tentang kinerja model dibandingkan dengan satu pembagian data (Kuhn, M., & Johnson, K. (2013).
   a. K-Fold Cross-Validation: Data dibagi menjadi K bagian yang sama, dan model dilatih dan diuji K kali, setiap kali menggunakan bagian yang berbeda sebagai data uji dan sisa bagian sebagai data latih.
   b. Leave-One-Out Cross-Validation (LOOCV): Versi ekstrem dari K-Fold Cross-Validation di mana setiap contoh data digunakan sebagai data uji satu kali dan sisanya sebagai data latih.
2. Split Data: Membagi dataset menjadi data latih dan data uji. Model dilatih pada data latih dan dievaluasi pada data uji yang tidak terlihat selama pelatihan.

3. Metrik Evaluasi:
   a. Akurasi: Proporsi prediksi yang benar dari semua prediksi. Digunakan untuk masalah klasifikasi.
   b. Precision, Recall, dan F1-Score: Metrik yang lebih terperinci untuk evaluasi klasifikasi, terutama ketika data tidak seimbang.
      1) Precision: Proporsi prediksi positif yang benar dari semua prediksi positif.
      2) Recall: Proporsi contoh positif yang benar yang ditemukan dari semua contoh positif.
      3) F1-Score: Rata-rata harmonis dari precision dan recall, memberikan keseimbangan antara keduanya.
   c. Mean Squared Error (MSE) dan Root Mean Squared Error (RMSE): Digunakan untuk masalah regresi, mengukur rata-rata kuadrat dari kesalahan prediksi dan akar kuadratnya.
   d. R^2 (Koefisien Determinasi): Mengukur seberapa baik model menjelaskan variasi dalam data target untuk regresi.

4. Kurva ROC dan AUC:
   a. ROC Curve (Receiver Operating Characteristic Curve): Menunjukkan trade-off antara true positive rate dan false positive rate untuk berbagai threshold.
   b. AUC (Area Under the Curve): Area di bawah kurva ROC, memberikan ukuran umum dari kemampuan model untuk membedakan antara kelas.

Daftar Pustaka

Bishop, C.M. (2006). Pattern Recognition and Machine Learning.

Breiman, L. (1996). Bagging Predictors. Machine Learning, 24(2), 123-140.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The Elements of Statistical Learning: Data Mining, Inference, and Prediction.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R. Springer.

Kotsiantis, S.B., Kanellopoulos, D., & Pintelas, P.E. (2006). Data Preprocessing for Supervised Learning. *International Journal of Computer Science*, 1(2), 111-120.**

Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling.

Quinlan, J.R. (1986). Induction of Decision Trees. *Machine Learning*, 1(1), 81-106.

Schölkopf, B., & Smola, A.J. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.

BAB 5

K-Medoids Clustering pada Machine Learning

-Andi Asari-

A. Konsep Dasar Clustering pada Machine Learning

Clustering adalah salah satu metode dalam machine learning yang digunakan untuk mengelompokkan data berdasarkan kemiripan atau kedekatan antar data. Clustering ini termasuk dalam jenis unsupervised learning, di mana model tidak memiliki label atau kategori yang sudah ditentukan sebelumnya untuk data yang dianalisis. Salah satu tujuan utama dari clustering adalah untuk menemukan struktur dalam data yang belum terstruktur, sehingga pola atau hubungan antar data dapat diidentifikasi secara otomatis.

1. Pengertian Clustering

Menurut Tan, Steinbach, dan Kumar (2018) Clustering adalah proses pengelompokan sekumpulan objek atau data ke dalam kelompok-kelompok sedemikian rupa sehingga objek-objek dalam satu kelompok

memiliki tingkat kemiripan yang tinggi satu sama lain dibandingkan dengan objek-objek dari kelompok yang lain.

Dalam konteks machine learning, clustering sering digunakan dalam berbagai aplikasi seperti analisis pemasaran, pengelompokan dokumen, segmentasi citra, dan bioinformatika. Misalnya, dalam pemasaran, clustering dapat digunakan untuk mengidentifikasi segmen pelanggan yang memiliki preferensi yang sama berdasarkan data pembelian mereka.

2. Algoritma Clustering

Terdapat berbagai macam algoritma clustering yang digunakan dalam machine learning, seperti K-Means, K-Medoids, DBSCAN, dan Hierarchical Clustering. Setiap algoritma memiliki karakteristik dan kegunaan yang berbeda tergantung pada jenis data dan tujuan analisis.

K-Means Clustering adalah salah satu algoritma clustering yang paling populer dan paling sering digunakan. Algoritma ini bekerja dengan membagi data ke dalam kelompok berdasarkan jarak rata-rata antar titik data di dalam setiap kelompok. Namun, K-Means memiliki beberapa kelemahan, seperti sensitifitas terhadap outlier dan kesulitan dalam menentukan jumlah cluster yang optimal.

Menurut Kaufman dan Rousseeuw (1990) K-Medoids Clustering adalah varian dari K-Means yang lebih robust terhadap outlier. Alih-alih menggunakan rata-rata dari semua data dalam satu cluster (seperti dalam K-Means), K-Medoids memilih satu titik data sebagai pusat dari cluster (medoid), yang meminimalkan

total jarak antara medoid dan semua titik data dalam cluster tersebut.

3. Aplikasi Clustering dalam Machine Learning

Clustering memiliki aplikasi yang luas dalam berbagai bidang. Salah satu contoh penggunaan clustering adalah dalam segmentasi pelanggan, di mana perusahaan dapat mengelompokkan pelanggan berdasarkan pola pembelian mereka untuk mengidentifikasi segmen yang berbeda dan merancang strategi pemasaran yang lebih tepat sasaran.

Dalam bidang bioinformatika, clustering digunakan untuk menganalisis data genetik dan mengelompokkan gen yang memiliki fungsi serupa atau yang berinteraksi dalam cara yang mirip. Clustering juga digunakan dalam pengolahan citra, di mana algoritma clustering digunakan untuk segmentasi citra dengan membagi citra menjadi bagian-bagian yang lebih kecil berdasarkan warna atau tekstur.

Sebagai contoh, dalam studi oleh Liu *et al.* (2021), clustering digunakan untuk mengelompokkan sel dalam data single-cell RNA sequencing, yang membantu dalam mengidentifikasi subpopulasi sel dengan ekspresi gen yang serupa. Penelitian ini menunjukkan bagaimana clustering dapat digunakan untuk mengungkap struktur yang kompleks dalam data biologis.

4. Tantangan dan Solusi dalam Clustering

Meskipun clustering adalah teknik yang sangat berguna, ada beberapa tantangan yang perlu diatasi. Salah satu tantangan utama adalah menentukan jumlah cluster yang optimal. Kebanyakan algoritma clustering,

seperti K-Means dan K-Medoids, memerlukan penentuan jumlah cluster sebagai input. Namun, dalam praktiknya, jumlah cluster yang optimal sering kali tidak diketahui sebelumnya dan dapat sangat mempengaruhi hasil clustering.

Untuk mengatasi masalah ini, berbagai metode telah dikembangkan, seperti penggunaan indeks validasi cluster, seperti Silhouette Score atau Davies-Bouldin Index, yang dapat membantu dalam menilai kualitas clustering dan menentukan jumlah cluster yang optimal.

Selain itu, kesulitan dalam menangani data dengan dimensi yang tinggi juga menjadi tantangan dalam clustering. Ketika dimensi data meningkat, jarak antara titik data menjadi kurang bermakna, yang dapat mempengaruhi kinerja algoritma clustering. Salah satu pendekatan untuk mengatasi masalah ini adalah dengan melakukan reduksi dimensi sebelum menerapkan algoritma clustering, seperti menggunakan Principal Component Analysis (PCA) atau t-Distributed Stochastic Neighbor Embedding (t-SNE).

B. K-Medoids Clustering pada Machine Learning

Dalam bidang pembelajaran mesin (machine learning), klastering adalah metode yang sering digunakan untuk mengelompokkan data ke dalam kelompok-kelompok berdasarkan kesamaan tertentu. Salah satu algoritma yang cukup populer dalam klastering adalah K-medoids. Algoritma ini merupakan variasi dari K-means, tetapi memiliki keunggulan dalam hal stabilitas terhadap outlier atau data yang menyimpang. K-medoids menggunakan salah satu titik data yang nyata sebagai pusat klaster, bukan centroid yang merupakan rata-rata dari titik-titik data dalam

klaster. Hal ini membuat K-medoids lebih robust dalam menghadapi data yang memiliki outlier.

1. Pengertian K-Medoids Clustering

K-medoids clustering adalah algoritma partisi yang membagi data menjadi kelompok, di mana setiap kelompok diwakili oleh salah satu objek dalam dataset yang berfungsi sebagai medoid. Menurut Kaufman dan Rousseeuw (1990), "Medoid adalah objek yang memiliki jarak rata-rata terkecil terhadap objek lain di dalam klasternya." Dengan kata lain, medoid adalah objek yang paling representatif dari kelompok tersebut.

Algoritma K-medoids bekerja dengan memilih secara acak medoid dari dataset sebagai pusat awal. Kemudian, algoritma ini mengelompokkan setiap objek ke medoid terdekat berdasarkan metrik jarak yang telah ditentukan, misalnya jarak Manhattan atau jarak Euclidean. Setelah semua objek dikelompokkan, medoid yang baru dipilih dengan tujuan meminimalkan total jarak dalam klaster. Proses ini diulang hingga tidak ada perubahan dalam penempatan medoid atau hingga mencapai iterasi maksimum.

2. Perbedaan K-Medoids dan K-Means

Meskipun K-medoids dan K-means sama-sama merupakan algoritma klastering yang bertujuan untuk membagi dataset menjadi kelompok, terdapat beberapa perbedaan penting antara keduanya. Pertama, K-means menggunakan rata-rata dari objek-objek dalam klaster sebagai centroid, sementara K-medoids menggunakan salah satu objek dalam klaster sebagai medoid. Hal ini menyebabkan K-medoids lebih tahan terhadap outlier, seperti yang diungkapkan dalam penelitian oleh Park dan

Jun (2009), "K-medoids lebih stabil dan robust terhadap outlier dibandingkan dengan K-means karena medoid selalu merupakan objek nyata dari dataset."

Kedua, K-means biasanya lebih cepat karena proses iterasinya hanya melibatkan perhitungan rata-rata, sementara K-medoids memerlukan perhitungan jarak antara setiap objek dalam klaster, yang lebih kompleks secara komputasi. Namun, untuk dataset yang mengandung banyak outlier atau memiliki distribusi yang tidak normal, K-medoids cenderung memberikan hasil yang lebih akurat.

3. Algoritma K-Medoids

Langkah-langkah dalam algoritma K-medoids adalah sebagai berikut:

a. Inisialisasi: Pilih secara acak objek dari dataset sebagai medoid awal.
b. Penugasan: Setiap objek dalam dataset ditugaskan ke medoid terdekat berdasarkan metrik jarak yang telah dipilih.
c. Pembaharuan Medoid: Untuk setiap klaster, pilih objek yang meminimalkan total jarak dalam klaster tersebut sebagai medoid baru.
d. Iterasi: Ulangi langkah penugasan dan pembaharuan medoid hingga tidak ada perubahan dalam medoid atau hingga mencapai iterasi maksimum. Proses iteratif ini terus berlanjut hingga tercapai kondisi konvergen, di mana medoid tidak lagi berubah. Pada titik ini, klaster dianggap stabil.

4. Implementasi dalam Pembelajaran Mesin

K-medoids sering digunakan dalam aplikasi di mana kestabilan dan ketahanan terhadap outlier adalah prioritas utama. Misalnya, dalam bidang bioinformatika, K-medoids digunakan untuk mengelompokkan data genetik yang seringkali mengandung banyak noise atau outlier. "K-medoids telah terbukti efektif dalam klastering data genetik karena kemampuannya untuk menangani outlier dengan lebih baik dibandingkan algoritma lain (Xu dan Wunsch 2005).

Di era big data, implementasi K-medoids juga dapat ditemukan dalam analisis perilaku konsumen, segmentasi pasar, dan penemuan pola dalam data besar yang memerlukan analisis robust. Algoritma ini sering diimplementasikan dengan bantuan perangkat lunak seperti R, Python (melalui library seperti Scikit-learn), atau MATLAB, yang menyediakan fungsi-fungsi untuk melakukan klastering menggunakan K-medoids.

5. Tantangan dan Solusi

Meskipun K-medoids menawarkan banyak keunggulan, penggunaannya tidak terlepas dari tantangan. Salah satu tantangan terbesar adalah kompleksitas komputasi yang lebih tinggi dibandingkan dengan K-means, terutama untuk dataset yang sangat besar. Untuk mengatasi masalah ini, beberapa pendekatan seperti PAM (Partitioning Around Medoids) dan algoritma CLARA (Clustering LARge Applications) telah dikembangkan. "CLARA adalah versi scalable dari K-medoids yang dirancang untuk menangani dataset besar dengan memilih sampel dari data dan menerapkan algoritma

PAM pada sampel tersebut (Kaufman dan Rousseeuw, 1990).

C. K-Medoids dalam Alur Kerja Machine Learning

Dalam lanskap machine learning, klastering adalah teknik yang penting untuk mengelompokkan data tanpa pengawasan (unsupervised). Salah satu metode yang menonjol dalam teknik ini adalah K-medoids, yang sering dibandingkan dengan K-means. Meskipun kedua metode ini beroperasi dengan tujuan yang sama, yakni meminimalkan jarak antara titik data dan pusat klaster, K-medoids memiliki pendekatan yang berbeda dalam menentukan pusat klaster tersebut.

K-medoids, atau yang juga dikenal sebagai Partitioning Around Medoids (PAM), merupakan algoritma klastering yang memanfaatkan medoid sebagai representasi pusat klaster. Berbeda dengan K-means yang menggunakan rata-rata titik data dalam klaster sebagai centroid, K-medoids memilih medoid, yaitu titik data yang benar-benar ada dalam klaster dan memiliki jarak total minimum terhadap semua titik lain dalam klaster tersebut. Hal ini membuat K-medoids lebih tahan terhadap pengaruh outlier dan data noise, yang sering kali menjadi kelemahan dalam algoritma K-means.

Menurut Kaufman dan Rousseeuw (1990), "Medoids lebih representatif sebagai pusat klaster dibandingkan centroid karena mereka adalah elemen aktual dari dataset, bukan hasil perhitungan rata-rata" (Kaufman & Rousseeuw, 1990). Pernyataan ini menekankan keunggulan K-medoids dalam situasi di mana integritas data sangat penting.

1. Proses Kerja K-Medoids dalam Machine Learning

Alur kerja K-medoids dalam machine learning dimulai dengan pemilihan sejumlah medoid secara acak dari dataset. Setelah medoid awal dipilih, setiap titik data yang tersisa diatribusikan ke klaster yang medoidnya paling dekat berdasarkan metrik jarak yang dipilih, seperti Manhattan distance atau Euclidean distance.

Proses ini kemudian diikuti dengan fase pertukaran (swap phase), di mana setiap medoid dicoba untuk diganti dengan titik non-medoid yang ada di klaster yang sama. Penggantian ini dilakukan hanya jika menghasilkan penurunan total jarak dalam klaster tersebut. Proses ini berlanjut hingga tidak ada perubahan lebih lanjut dalam medoid atau penurunan jarak.

Misal, jika kita memiliki dataset dengan tiga klaster, kita akan memulai dengan memilih tiga titik acak sebagai medoid. Selanjutnya, titik-titik lainnya dikelompokkan berdasarkan kedekatannya dengan salah satu dari tiga medoid tersebut. Setelah itu, kita mencoba mengganti medoid yang sudah dipilih dengan titik lain dalam klaster untuk melihat apakah total jarak dapat diminimalkan. Algoritma ini akan berhenti ketika tidak ada medoid yang bisa diganti lagi tanpa meningkatkan total jarak dalam klaster.

Menurut Park dan Jun (2009), "Proses iteratif dalam K-medoids memastikan bahwa solusi akhir lebih stabil dibandingkan K-means, terutama dalam kehadiran data anomali" (Park & Jun, 2009). Ini menunjukkan keandalan K-medoids dalam skenario dunia nyata di mana data sering kali tidak sempurna.

2. Keunggulan dan Kelemahan K-Medoids

Keunggulan utama K-medoids adalah ketahanannya terhadap outlier dan noise, yang sering kali memengaruhi hasil klastering dalam K-means. Medoid yang digunakan dalam K-medoids adalah titik aktual dari dataset, sehingga tidak terpengaruh oleh nilai ekstrem yang mungkin mempengaruhi perhitungan rata-rata dalam K-means. Selain itu, algoritma ini cenderung lebih stabil karena hasil akhirnya tidak bergantung pada pemilihan medoid awal yang acak.

Namun, K-medoids juga memiliki beberapa kelemahan. Pertama, algoritma ini lebih lambat dibandingkan K-means, terutama pada dataset yang besar, karena setiap iterasi memerlukan perhitungan jarak antara semua titik data dalam klaster dengan medoid yang baru. Kedua, algoritma ini kurang efisien dalam hal skala, sehingga memerlukan optimasi lebih lanjut ketika digunakan dalam big data.

Menurut Schubert *et al*. (2017), "Meskipun K-medoids memberikan hasil klastering yang lebih akurat pada data dengan outlier, biayanya adalah waktu komputasi yang lebih tinggi, terutama pada dataset besar" (Schubert *et al*., 2017). Hal ini menunjukkan pentingnya mempertimbangkan ukuran dataset ketika memilih antara K-means dan K-medoids.

3. Aplikasi K-Medoids dalam Machine Learning

K-medoids memiliki berbagai aplikasi dalam machine learning, terutama dalam situasi di mana data memiliki outlier atau distribusi yang tidak merata. Salah satu aplikasi utama adalah dalam pengelompokan dokumen, di mana dokumen-dokumen diorganisasikan

ke dalam kelompok-kelompok berdasarkan kemiripan isinya. Dalam konteks ini, K-medoids dapat membantu mengidentifikasi topik utama dalam kumpulan dokumen yang besar, yang sering kali mengandung dokumen-dokumen yang berbeda jauh dari topik utama. Contoh lain adalah dalam analisis pasar, di mana K-medoids digunakan untuk mengelompokkan pelanggan berdasarkan pola pembelian mereka. Dengan mengidentifikasi medoid sebagai pelanggan yang paling representatif dalam setiap klaster, bisnis dapat menargetkan kampanye pemasaran mereka lebih efektif.

Menurut Hastie, Tibshirani, dan Friedman (2009), "Penggunaan K-medoids dalam analisis pasar memungkinkan perusahaan untuk memahami lebih baik kebutuhan dan preferensi pelanggan, yang pada gilirannya meningkatkan strategi pemasaran" (Hastie, Tibshirani, & Friedman, 2009). Hal ini menunjukkan potensi K-medoids dalam memberikan wawasan yang lebih mendalam dibandingkan teknik klastering lainnya.

4. Perbandingan dengan Algoritma Klastering Lainnya

K-medoids sering kali dibandingkan dengan algoritma klastering lainnya, seperti K-means, DBSCAN, dan hierarchical clustering. Meskipun setiap algoritma memiliki keunggulan dan kelemahan masing-masing, pilihan algoritma sering kali bergantung pada karakteristik data dan tujuan klastering. K-means, misalnya, lebih efisien dari segi komputasi dan cocok untuk data dengan distribusi yang lebih seragam dan tanpa outlier. Di sisi lain, DBSCAN lebih baik dalam mengidentifikasi klaster dengan bentuk yang tidak beraturan dan dapat menangani noise dengan lebih baik.

Hierarchical clustering menawarkan fleksibilitas dalam memilih jumlah klaster setelah proses klastering selesai, namun cenderung lebih lambat dan tidak skala pada dataset besar.

Menurut Berkhin (2006), "Pemilihan algoritma klastering yang tepat harus mempertimbangkan sifat data dan tujuan analisis, dengan K-medoids menjadi pilihan yang baik ketika keakuratan dan ketahanan terhadap outlier menjadi prioritas utama" (Berkhin, 2006). Pernyataan ini menekankan pentingnya pemahaman karakteristik data sebelum memilih algoritma klastering.

5. Implementasi K-Medoids dalam Alur Kerja Machine Learning

Implementasi K-medoids dalam alur kerja machine learning melibatkan beberapa langkah kunci, mulai dari pra-pemrosesan data, pemilihan medoid awal, hingga evaluasi hasil klastering. Pra-pemrosesan data adalah langkah pertama yang penting, termasuk normalisasi atau standarisasi data untuk memastikan bahwa semua fitur memiliki skala yang sama. Setelah data siap, pemilihan medoid awal dilakukan, biasanya secara acak. Namun, pemilihan medoid yang cermat dapat meningkatkan kecepatan konvergensi algoritma. Setelah medoid dipilih, data diklasterkan berdasarkan kedekatannya dengan medoid, dan proses pertukaran medoid dilakukan untuk meminimalkan total jarak dalam klaster. Langkah terakhir adalah evaluasi hasil klastering, yang dapat dilakukan dengan metrik seperti silhouette score atau Davies-Bouldin Index. Evaluasi ini penting untuk menentukan seberapa baik klaster telah terbentuk dan

apakah hasil klastering dapat memberikan wawasan yang berguna.

Menurut Xu dan Wunsch (2005), "Evaluasi hasil klastering sangat penting untuk memastikan bahwa klaster yang dihasilkan tidak hanya sesuai dengan data tetapi juga memberikan nilai yang berarti bagi analisis lebih lanjut" (Xu & Wunsch, 2005). Ini menunjukkan pentingnya evaluasi yang menyeluruh dalam alur kerja machine learning.

6. Tantangan dan Solusi dalam Penerapan K-Medoids

Meskipun K-medoids menawarkan banyak keunggulan, penerapannya juga dihadapkan pada beberapa tantangan, terutama terkait dengan skala dan efisiensi komputasi. Algoritma ini cenderung lambat pada dataset besar karena setiap iterasi melibatkan perhitungan jarak antara setiap titik data dan medoid yang potensial. Selain itu, pemilihan medoid awal yang buruk dapat mengakibatkan hasil klastering yang suboptimal. Untuk mengatasi tantangan ini, beberapa pendekatan telah dikembangkan. Salah satu pendekatan adalah penggunaan teknik sampling, di mana hanya sebagian data yang digunakan untuk menentukan medoid awal, yang kemudian diperbaiki pada iterasi berikutnya. Pendekatan lainnya adalah penggunaan metode optimasi seperti simulated annealing atau genetic algorithms untuk menemukan medoid yang lebih baik.

Menurut Arthur dan Vassilvitskii (2007), "Penggunaan metode optimasi dalam K-medoids dapat secara signifikan meningkatkan hasil klastering, terutama dalam situasi di mana data memiliki struktur yang kompleks" (Arthur & Vassilvitskii, 2007). Ini menunjukkan bahwa

tantangan dalam K-medoids dapat diatasi dengan strategi yang tepat.

D. Implementasi K-medoids Clustering

K-medoids adalah algoritma klastering yang efektif dalam machine learning dan analisis data. Algoritma ini sering dipilih karena kemampuannya untuk mengelompokkan data dengan baik, terutama ketika data mengandung outlier atau noise. K-medoids, yang juga dikenal sebagai Partitioning Around Medoids (PAM), memilih medoid sebagai pusat klaster. Medoid adalah titik yang benar-benar ada dalam dataset dan memiliki jarak total minimum ke semua titik lain dalam klaster tersebut.

1. Proses Implementasi K-medoids

Implementasi K-medoids dapat diuraikan dalam beberapa langkah kunci:

a. Inisialisasi Medoid: Pilih sejumlah medoid awal secara acak dari dataset. Jumlah medoid ini adalah parameter K yang menentukan jumlah klaster yang akan dibentuk.
b. Pengelompokan Data: Alokasikan setiap titik data ke medoid terdekat berdasarkan metrik jarak yang dipilih, seperti Euclidean distance atau Manhattan distance.
c. Optimasi Medoid: Iterasi melalui proses pertukaran (swap) di mana setiap medoid diuji untuk diganti dengan titik non-medoid dalam klaster yang sama. Jika penggantian medoid menghasilkan penurunan total jarak, maka penggantian dilakukan.
d. Konvergensi: Proses ini berlanjut hingga tidak ada perubahan signifikan dalam medoid, atau jarak total tidak dapat diminimalkan lebih lanjut.

Misalnya, dalam dataset dengan 1000 titik data dan 5 klaster yang diinginkan, algoritma K-medoids akan memulai dengan memilih 5 titik acak sebagai medoid. Setiap titik data kemudian akan dikelompokkan berdasarkan kedekatannya dengan medoid. Setelah setiap iterasi, medoid yang lebih baik akan dicari, dan proses ini akan diulang hingga stabil.

Menurut Park dan Jun (2009), "Proses pertukaran medoid dalam K-medoids memastikan bahwa pusat klaster yang dipilih adalah yang paling representatif, sehingga menghasilkan klaster yang lebih stabil" (Park & Jun, 2009). Ini menunjukkan bahwa pemilihan medoid yang tepat penting untuk kualitas hasil klastering.

2. Keunggulan K-medoids

K-medoids memiliki beberapa keunggulan dibandingkan dengan metode klastering lainnya:

a. Ketahanan Terhadap Outlier: Karena medoid adalah titik data aktual, K-medoids tidak terpengaruh oleh outlier dan nilai ekstrem yang dapat memengaruhi centroid dalam metode K-means.
b. Hasil Klastering yang Stabil: K-medoids cenderung menghasilkan hasil yang lebih stabil karena tidak bergantung pada rata-rata titik data dalam klaster.
c. Tidak Memerlukan Asumsi Distribusi: Berbeda dengan beberapa metode klastering yang mengasumsikan distribusi data tertentu, K-medoids tidak memerlukan asumsi tersebut, sehingga lebih fleksibel untuk berbagai jenis data.

Menurut Schubert *et al.* (2017), "Keuntungan utama K-medoids adalah ketahanannya terhadap data yang

tidak bersih dan outlier, yang sering menjadi masalah dalam metode klastering lainnya" (Schubert *et al.,* 2017). Ini menunjukkan bahwa K-medoids sangat berguna dalam aplikasi dunia nyata di mana data sering kali tidak sempurna.

3. Kelemahan K-medoids

Meskipun K-medoids memiliki banyak keunggulan, ada beberapa kelemahan yang perlu diperhatikan:

a. Kinerja pada Dataset Besar: K-medoids dapat menjadi lambat pada dataset besar karena perhitungan jarak antara semua titik data dan medoid yang potensial pada setiap iterasi.
b. Pemilihan Medoid Awal: Pemilihan medoid awal yang buruk dapat memengaruhi hasil klastering. Proses ini sering kali memerlukan teknik inisialisasi yang cermat untuk menghindari hasil yang suboptimal.
c. Keterbatasan Skalabilitas: Algoritma ini kurang efisien dalam hal skala, sehingga mungkin memerlukan optimasi tambahan untuk diterapkan pada big data.

Menurut Xu dan Wunsch (2005), "Kelemahan utama K-medoids adalah waktu komputasi yang tinggi pada dataset besar, yang memerlukan teknik optimasi untuk meningkatkan efisiensi" (Xu & Wunsch, 2005). Ini menunjukkan bahwa meskipun K-medoids efektif, efisiensi komputasi tetap menjadi tantangan utama.

4. Aplikasi K-medoids dalam Berbagai Bidang

K-medoids memiliki aplikasi luas dalam berbagai bidang, termasuk:

a. Pengelompokan Dokumen: Dalam analisis teks, K-medoids digunakan untuk mengelompokkan dokumen berdasarkan kemiripan kontennya. Hal ini memungkinkan identifikasi topik utama dan pengorganisasian dokumen yang lebih efisien.
b. Segmentasi Pasar: K-medoids dapat membantu dalam segmentasi pasar dengan mengelompokkan pelanggan berdasarkan pola pembelian atau preferensi mereka. Ini memungkinkan strategi pemasaran yang lebih terarah.
c. Analisis Genetik: Dalam bidang bioinformatika, K-medoids digunakan untuk mengelompokkan data genetik berdasarkan ekspresi gen. Ini membantu dalam pemahaman pola genetik dan identifikasi kelompok gen yang relevan.

Menurut Hastie, Tibshirani, dan Friedman (2009), "K-medoids sangat berguna dalam pengelompokan data yang kompleks dan bervariasi, seperti dalam analisis genetik dan segmentasi pasar" (Hastie, Tibshirani, & Friedman, 2009). Ini menunjukkan fleksibilitas K-medoids dalam berbagai aplikasi analisis data.

5. Perbandingan dengan Metode Klastering Lainnya

K-medoids sering dibandingkan dengan metode klastering lainnya, seperti K-means, DBSCAN, dan hierarchical clustering. Setiap metode memiliki kelebihan dan kekurangan masing-masing:

a. K-means: Lebih efisien dalam hal komputasi dan cocok untuk data dengan distribusi yang lebih seragam. Namun, sensitif terhadap outlier dan noise.

b. DBSCAN: Baik dalam mengidentifikasi klaster dengan bentuk yang tidak beraturan dan menangani noise. Namun, memerlukan parameter yang tepat untuk menentukan radius dan jumlah titik minimum dalam klaster.
c. Hierarchical Clustering: Menyediakan fleksibilitas dalam menentukan jumlah klaster setelah proses klastering. Namun, cenderung lebih lambat dan kurang efisien pada dataset besar.

Menurut Berkhin (2006), "Pemilihan algoritma klastering harus didasarkan pada karakteristik data dan tujuan analisis, dengan K-medoids menjadi pilihan yang baik ketika ketahanan terhadap outlier dan hasil yang stabil sangat penting" (Berkhin, 2006). Ini menunjukkan bahwa pemilihan metode klastering harus mempertimbangkan berbagai faktor untuk mendapatkan hasil yang optimal.

6. Tantangan dan Solusi dalam Penerapan K-medoids

Beberapa tantangan dalam penerapan K-medoids meliputi:

a. Waktu Komputasi: Untuk mengatasi waktu komputasi yang tinggi, terutama pada dataset besar, teknik optimasi seperti sampling dan penggunaan algoritma metaheuristik (simulated annealing, genetic algorithms) dapat diterapkan.
b. Pemilihan Medoid Awal: Teknik inisialisasi seperti K-medoids++ dapat digunakan untuk meningkatkan pemilihan medoid awal dan mengurangi kemungkinan hasil suboptimal.

c. Scalability: Untuk meningkatkan skalabilitas, teknik optimasi paralel dan distribusi dapat diterapkan untuk mempercepat proses klastering pada dataset besar.

Menurut Arthur dan Vassilvitskii (2007), "Penggunaan teknik optimasi dan inisialisasi yang cermat dapat mengatasi banyak kelemahan K-medoids, terutama dalam hal waktu komputasi dan kualitas hasil" (Arthur & Vassilvitskii, 2007). Ini menunjukkan bahwa tantangan dalam K-medoids dapat diatasi dengan pendekatan yang tepat.

E. Tantangan dan Arahan Masa Depan K-medoids Clustering

K-medoids clustering adalah metode yang efektif dalam analisis data dan machine learning, terutama ketika menghadapi data yang kompleks dan noisy. Meskipun K-medoids memiliki banyak keunggulan, seperti ketahanan terhadap outlier dan stabilitas dalam klastering, metode ini juga menghadapi berbagai tantangan dalam implementasinya. Bab ini akan membahas tantangan-tantangan utama yang dihadapi oleh K-medoids dan mengeksplorasi arahan masa depan untuk mengatasi isu-isu tersebut serta meningkatkan kinerja algoritma.

1. Tantangan dalam Implementasi K-medoids

a. Waktu Komputasi dan Skalabilitas

Salah satu tantangan terbesar dalam penerapan K-medoids adalah waktu komputasi yang tinggi, terutama pada dataset besar. Proses yang melibatkan pemilihan medoid yang optimal melalui pertukaran (*swap*) memerlukan waktu yang signifikan, yang dapat menjadi tidak praktis pada skala besar.

Menurut Xu dan Wunsch (2005), "Algoritma K-medoids secara signifikan lebih lambat dibandingkan dengan K-means karena kebutuhan untuk menghitung jarak antara semua pasangan titik, yang dapat menjadi sangat mahal pada dataset besar" (Xu & Wunsch, 2005). Oleh karena itu, peningkatan efisiensi algoritma K-medoids adalah fokus utama dalam penelitian dan pengembangan.

b. Inisialisasi Medoid

Pemilihan medoid awal mempengaruhi hasil klastering dan kualitas solusi akhir. Inisialisasi medoid yang buruk dapat menyebabkan konvergensi ke solusi lokal yang suboptimal. Hal ini menggarisbawahi kebutuhan untuk teknik inisialisasi yang lebih baik.

Menurut Arthur dan Vassilvitskii (2007), "Kualitas hasil klastering sangat bergantung pada pemilihan medoid awal, dan penggunaan metode inisialisasi yang lebih baik seperti K-medoids++ dapat meningkatkan hasil klastering" (Arthur & Vassilvitskii, 2007). Teknik seperti K-medoids++ dan pendekatan berbasis heuristik dapat membantu mengatasi masalah ini.

c. Keterbatasan dalam Menangani Data yang Sangat Tidak Seimbang

K-medoids mungkin tidak selalu efektif dalam menangani dataset yang sangat tidak seimbang. Ketika ada perbedaan besar dalam ukuran klaster, algoritma dapat menghasilkan hasil yang tidak memadai.

Menurut Park dan Jun (2009), "Ketidakseimbangan dalam ukuran klaster dapat menyebabkan

kesulitan dalam menemukan medoid yang representatif, yang pada gilirannya memengaruhi kualitas klastering" (Park & Jun, 2009). Penelitian lebih lanjut diperlukan untuk mengembangkan teknik yang dapat menangani ketidakseimbangan data dengan lebih baik.

d. Sensitivitas Terhadap Jarak dan Metode Pengukuran

K-medoids sangat bergantung pada metrik jarak yang digunakan, seperti Euclidean atau Manhattan distance. Pilihan metrik yang tidak sesuai dapat mempengaruhi hasil klastering secara signifikan.

Menurut Schubert *et al.* (2017), "Pemilihan metrik jarak yang tepat sangat penting untuk algoritma K-medoids, dan ketidakcocokan dalam pemilihan metrik dapat mengakibatkan hasil klastering yang tidak optimal" (Schubert *et al.,* 2017). Oleh karena itu, eksplorasi metrik jarak alternatif dan adaptif menjadi area penelitian yang penting.

2. Arahan Masa Depan untuk K-medoids

a. Teknik Optimasi dan Peningkatan Kinerja

Untuk mengatasi tantangan waktu komputasi, beberapa teknik optimasi telah dikembangkan, termasuk penggunaan algoritma metaheuristik seperti simulated annealing dan algoritma genetika. Teknik-teknik ini dapat mempercepat proses pencarian medoid yang optimal dengan cara yang lebih efisien.

Menurut Zhou *et al.* (2018), "Algoritma metaheuristik seperti simulated annealing dan algoritma genetika dapat digunakan untuk meningkatkan efisiensi algoritma K-medoids dengan mengurangi jumlah iterasi yang diperlukan untuk

menemukan medoid yang optimal" (Zhou *et al.,* 2018). Penerapan teknik ini dapat membantu mempercepat proses klastering dan meningkatkan kinerja algoritma.

b. Pengembangan Teknik Inisialisasi yang Lebih Baik

Inisialisasi medoid yang lebih baik adalah area penting untuk penelitian lebih lanjut. Teknik seperti K-medoids++ dan metode berbasis heuristik dapat membantu memilih medoid awal yang lebih representatif dan mengurangi risiko konvergensi ke solusi lokal yang buruk.

Menurut Arthur dan Vassilvitskii (2007), "K-medoids++ adalah salah satu teknik inisialisasi yang menjanjikan, yang mengoptimalkan pemilihan medoid awal dan meningkatkan kualitas hasil klastering" (Arthur & Vassilvitskii, 2007). Penelitian lebih lanjut tentang teknik inisialisasi dapat memberikan solusi yang lebih efektif untuk masalah ini.

c. Penanganan Data Tidak Seimbang dan Skala Besar

Pengembangan metode untuk menangani data yang sangat tidak seimbang dan skala besar merupakan arah penting untuk masa depan K-medoids. Teknik seperti sampling data dan algoritma paralel dapat membantu dalam mengatasi masalah ini.

Menurut Xu dan Wunsch (2005), "Pendekatan seperti sampling data dan algoritma paralel dapat diterapkan untuk meningkatkan skalabilitas K-medoids dan mengatasi ketidakseimbangan data" (Xu & Wunsch, 2005). Teknik-teknik ini dapat membantu K-medoids beradaptasi dengan dataset besar dan tidak seimbang.

d. Eksplorasi Metrik Jarak Alternatif

Eksplorasi metrik jarak alternatif yang dapat beradaptasi dengan jenis data yang berbeda adalah area penelitian yang menjanjikan. Metrik jarak yang lebih fleksibel dapat meningkatkan efektivitas K-medoids dalam klastering.

Menurut Schubert *et al.* (2017), "Metrik jarak alternatif, seperti jarak berbasis kernel atau jarak berbasis distribusi, dapat memberikan hasil yang lebih baik dalam situasi di mana metrik standar tidak memadai" (Schubert et al., 2017). Penelitian lebih lanjut dalam area ini dapat memberikan wawasan baru tentang cara mengadaptasi K-medoids untuk berbagai jenis data.

3. Aplikasi dan Studi Kasus

a. Pengelompokan Data Genetik

Dalam bidang bioinformatika, K-medoids digunakan untuk mengelompokkan data genetik berdasarkan ekspresi gen. Penelitian oleh Rauschenberger *et al.* (2016) menunjukkan bahwa K-medoids dapat membantu mengidentifikasi pola ekspresi gen yang relevan dan meningkatkan pemahaman tentang kelompok gen (Rauschenberger *et al.*, 2016).

b. Segmentasi Pasar

Dalam pemasaran, K-medoids digunakan untuk segmentasi pasar dengan mengelompokkan pelanggan berdasarkan pola pembelian. Penelitian oleh Mazzarol *et al.* (2018) menunjukkan bahwa K-medoids dapat meningkatkan strategi pemasaran dengan meng-

identifikasi segmen pasar yang lebih relevan (Mazzarol *et al.*, 2018).

c. Analisis Dokumen

Dalam analisis teks, K-medoids digunakan untuk mengelompokkan dokumen berdasarkan kemiripan konten. Penelitian oleh Kamel *et al.* (2019) menunjukkan bahwa K-medoids dapat meningkatkan organisasi dan pengelompokan dokumen (Kamel *et al.*, 2019).

F. Kesimpulan

K-medoids clustering adalah metode klastering yang efektif, namun menghadapi berbagai tantangan seperti waktu komputasi, inisialisasi medoid, penanganan data tidak seimbang, dan sensitivitas terhadap metrik jarak. Untuk mengatasi tantangan ini, arahan masa depan mencakup teknik optimasi, pengembangan teknik inisialisasi yang lebih baik, penanganan data besar dan tidak seimbang, serta eksplorasi metrik jarak alternatif.

Implementasi dan penelitian lebih lanjut dalam area ini dapat meningkatkan kinerja K-medoids dan memperluas aplikasinya dalam berbagai bidang analisis data. Dengan pendekatan yang tepat, K-medoids dapat terus menjadi alat yang berguna dan efektif dalam machine learning dan analisis data.

Daftar Pustaka

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The Advantages of Careful Seeding. *In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA '07).

Berkhin, P. (2006). A Survey of Clustering Data Mining Techniques. *In Grouping Multidimensional Data*. Springer, Berlin, Heidelberg.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Kaufman, L., & Rousseeuw, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley.

Liu, Y., Zhang, X., Wang, J., & Li, Z. (2021). Single-cell RNA sequencing analysis reveals cell heterogeneity and gene expression dynamics in early-stage lung adenocarcinoma. *Nature Communications*, 12(1), 2541.

Park, H. S., & Jun, C. H. (2009). A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications,* 36(2), 3336-3341.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS),* 42(3), 19.

Tan, P.-N., Steinbach, M., & Kumar, V. (2018). Introduction to Data Mining. Pearson.

Xu, R., & Wunsch, D. (2005). Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks*, 16(3), 645-678.

Tentang Penulis

Panji Bintoro, S.Kom., M.Cs.
Dosen Rekayasa Perangkat Lunak
Fakultas Teknologi dan Informatika Universitas Aisyah Pringsewu

Penulis lahir di Gading Rejo tanggal 12 Desember 1996. Penulis adalah dosen pada Program Studi Rekayasa Perangkat Lunak Fakultas Teknologi dan Informatika, Universitas Aisyah Pringsewu. Menyelesaikan pendidikan S1 pada Jurusan Teknik Informatika, Universitas Ahmad Dahlan Yogyakarta dan melanjutkan S2 pada Jurusan Magister Ilmu Komputer, Universitas Gadjah Mada. Pria yang kerap disapa Panji ini adalah anak dari pasangan Marjito (ayah) dan Tri Pandu Ratsih (ibu).

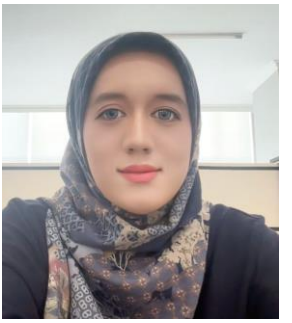
Ratnasari, S.Kom., M.Kom.
Dosen Rekayasa Perangkat Lunak
Fakultas Teknologi dan Informatika, Universitas Aisyah Pringsewu

Penulis lahir di Lampung pada tanggal 28 Juni 1993. Penulis adalah dosen tetap pada Program Studi Rekayasa Perangkata Lunak Fakultas Fakultas Teknologi & Informatika, Universitas Aisyah Pringsewu. Menyelesaikan pendidikan S1 Jurusan Teknik Informatika tahun 2015 di Universitas Indraprasta PGRI Jarakarta dan menyelesaikan S2 Jurusan Ilmu Komputer di Universitas Budiluhur Jakarta pada tahun 2018. Penulis menekuni bidang Menulis yang bertujuan untuk berbagi pengetahuan dan pengalaman kepada para mahasiswa dan profesional di bidang teknologi informasi. Penulis percaya bahwa dengan berbagi ilmu, bisa mendorong inovasi dan meningkatkan kualitas penerapan algoritma pembelajaran mesin dalam pengembangan perangkat lunak di Indonesia. Selain mengajar dan menulis buku, penulis juga aktif memberikan pelatihan dan workshop di berbagai perusahaan teknologi serta menjadi pembicara di konferensi nasional dan internasional.

Edy Wihardjo, S.Pd., M.Pd., MCE., MCF.
Dosen Pendidikan Matematika
PMIPA, FKIP, Universitas Jember

Lahir di Bondowoso pada tanggal 8 Januari, penulis adalah seorang dosen yang berdedikasi pada Program Studi Pendidikan Matematika, PMIPA, FKIP, Universitas Jember. Setelah meraih gelar Sarjana Pendidikan di Universitas Jember dan Magister Pendidikan di Universitas Negeri Malang, penulis melanjutkan studi ke Program Doktor di Universitas Negeri Surabaya.

Sebagai peneliti dan pengembang dalam pembelajaran berbasis teknologi informasi, khususnya di bidang matematika, penulis telah memberikan kontribusi yang signifikan dalam dunia pendidikan. Dedikasinya pada inovasi pembelajaran tidak hanya terbatas di ruang kelas tetapi juga tercermin dalam berbagai platform digital yang dikelola sesuai kepakarannya di bidang MAT (Matematika, Asesmen, dan Teknologi). Penulis aktif mengelola blog pembelajaran di https://mat.or.id, laman Facebook di https://fb.me/matemania2, akun Instagram di https://www.instagram.com/mat.or.id/, dan kanal YouTube di https://www.youtube.com/@pakarti. Melalui platform-platform ini, penulis berbagi wawasan, metode pengajaran, dan inovasi terkini yang dapat diakses oleh kalangan pendidik, (maha)siswa, dan masyarakat luas.



Indah Pratiwi Putri, B.IT.(Hons.), M.Sc.IT.
Dosen Sistem Informasi
Fakultas Ilmu Komputer dan Sains
Universitas Indo Global Mandiri Palembang

Penulis lahir di Palembang tanggal 30 Oktober 1988. Penulis adalah dosen tetap pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer dan Sains, Universitas Indo Global Mandiri Palembang. Penulis menyelesaikan pendidikan S1 pada Jurusan Information Technology dan melanjutkan S2 pada Information Technology di Universiti Utara Malaysia, Sintok, Kedah Darul Aman.

Penulis menekuni bidang penelitian machine learning, analisis data, kecerdasan buatan, dan pengembangan perangkat lunak, dengan fokus pada penerapan teknologi informasi dalam bidang akademis dan masyarakat. Penulis juga aktif terjun langsung mengabdi kepada Masyarakat dengan mengadakan acara pelatihan/bimtek/penyuluhan pemanfaatan teknologi informasi.

Penulis juga memiliki pengalaman kerja tiga belas tahun di industri pengembangan perangkat lunak, keamanan siber dan institusi pendidikan. Karir profesional penulis mencakup IT Business, System Analyst, Game Tester, ERP dan SAP Consultant.

Dengan latar belakang yang kuat dalam teknologi informasi dan pengalaman profesional yang luas, penulis berkomitmen untuk terus berkontribusi dalam pengembangan teknologi dan aplikasinya untuk mendukung pendidikan dan peningkatan kualitas hidup masyarakat.

Andi Asari, SIP., S.Kom., M.A.
Dosen Perpustakaan dan Sains Informasi
Fakultas Sastra

Penulis adalah dosen di Universitas Negeri Malang yang saat ini sedang melanjutkan studi doctoral (S3) di jurusan Information Management UiTM Malaysia. Penulis merupakan alumni dari Magister Kajian Budaya dan Media sekolah pasca sarjana Universitas Gadjah Mada Yogyakarta, dan juga alumni dari jurusan Ilmu Perpustakaan UIN Sunan Kalijaga Yogyakarta, serta alumni jurusan Teknik Informatika STMIK.

Mulai tahun 2015 sampai sekarang penulis aktif mengajar di Jurusan Sastra Indonesia, Prodi S1 Ilmu Perpustakaan dan D4 Perpustakaan Digital Universitas Negeri Malang. Disamping kesibukan di dunia akademis juga memiliki kegiatan sebagai narasumber pada kegiatan seminar, workshop, konsultan lembaga pendidikan dan perpustakaan.