Pada era digital ini, *machine learning* telah menjadi salah satu topik yang paling menarik untuk dibahas. Dari aplikasi canggih yang membantu kita dalam kehidupan sehari-hari hingga algoritma kompleks yang mendorong inovasi di berbagai industri, dunia *artificial intelligence* (AI) menawarkan banyak sekali potensi.

Namun, di balik setiap model AI yang sukses, terdapat serangkaian langkah terstruktur yang dikenal sebagai ***machine learning workflow***. Mungkin kamu sudah pernah mendengar istilah ini, tetapi bagaimana cara kita memastikan bahwa sudah mengikuti proses yang benar untuk membangun model yang efektif?

Dengan begitu banyak informasi yang berseliweran di internet, penting untuk memiliki *workflow* yang jelas agar tidak tersesat di tengah banyaknya konsep yang ada.

Workflow ini terdiri dari berbagai tahap yang saling berkaitan, mulai dari mendefinisikan masalah hingga melakukan pemeliharaan model setelah tahapan *deployment*. Memahami setiap langkah ini bukan hanya membantu kamu dalam membangun model yang lebih baik, tetapi juga memberikan gambaran menyeluruh tentang perjalanan data dari awal hingga akhir.

Jadi, siapkah kamu menjelajahi langkah-langkah praktis untuk membangun model AI yang canggih? Dalam blog ini, kita akan membahas setiap tahap dalam machine learning workflow dengan cara yang mudah dipahami. Jangan lupa, jika kamu punya pengalaman atau pertanyaan, bagikan di kolom komentar, ya. Yuk, kita mulai perjalanan ini bersama-sama! 🚀

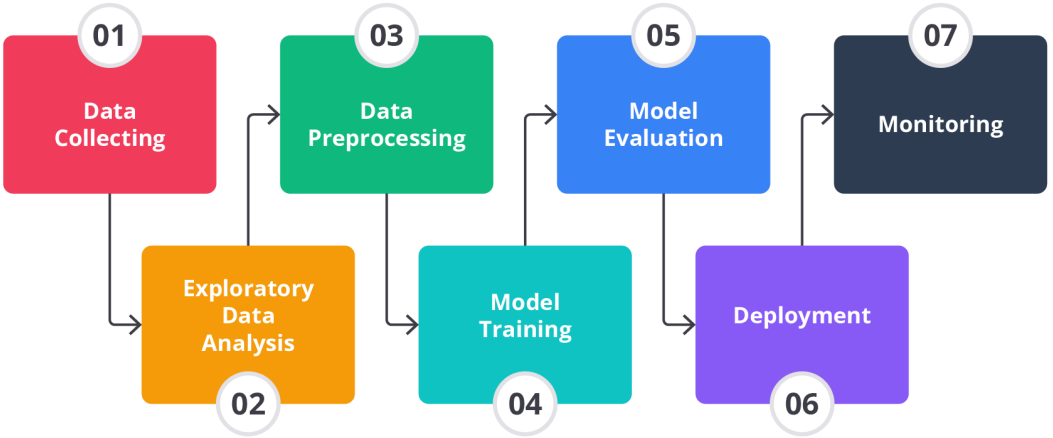
Pendahuluan

Tahukah kamu bahwa setiap kali kita bepergian, biasanya kita akan mempersiapkan rute, *entah* itu dengan peta konvensional atau peta digital? Yang menarik, saat kita sudah mahir dengan rute tertentu, kita bahkan bisa memberi tahu orang lain tentang jalan yang harus mereka ambil. Keren, *kan?*

Nah, mari kita berbicara tentang fungsi peta. Dalam konteks perjalanan, peta sangat penting dan simbolis. Sebagai panduan dan alat navigasi, peta memberikan arah yang jelas dan membantu kita merencanakan rute terbaik agar bisa mencapai tujuan dengan efisien dan aman.

Hal ini juga berlaku ketika membangun proyek machine learning. Kita butuh panduan yang jelas untuk merancang model terbaik. Di sinilah machine learning workflow berperan. Workflow ini berisi tahapan-tahapan yang perlu dilalui sebelum proyek kita bisa diterapkan di dunia nyata.

Aurélien Géron, dalam bukunya "Hands-On Machine Learning with Scikit-Learn & TensorFlow," menjelaskan bahwa machine learning workflow terdiri dari **langkah-langkah sistematis** yang dimulai dari mendefinisikan masalah hingga menerapkan model. Jika kita rangkum, hasilnya bisa digambarkan dalam sebuah diagram yang menggambarkan proses tersebut.



Alur kerja machine learning bersifat sistematis dan iteratif, yang artinya setiap langkah perlu dievaluasi dan disesuaikan untuk memastikan model yang kita hasilkan akurat dan dapat diandalkan di lingkungan produksi. Buku ini juga menekankan betapa pentingnya memahami data dan masalah yang dihadapi, serta menggunakan alat dan teknik yang tepat untuk mencapai tujuan machine learning kita.

Melihat diagram di atas, mungkin kamu berpikir bahwa ada banyak tahapannya, ya? Namun, jangan khawatir! Kita akan mempelajari semua langkah ini dari nol, dengan harapan bisa memperkaya pengetahuan dasar yang akan sangat membantu dalam proyek-proyek kamu di masa depan.

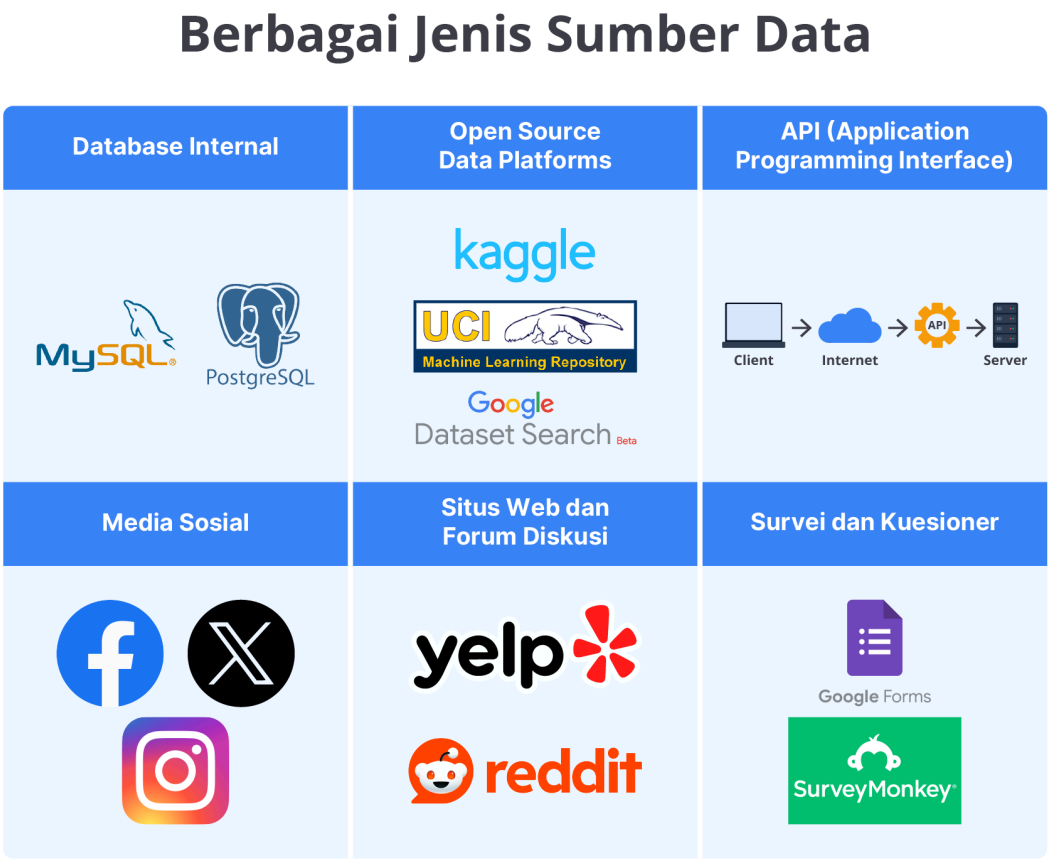
Tanpa menunggu lebih lama, mari kita lihat lebih dekat apa yang tergambar dalam diagram di atas!

1. Data Collecting atau Pengumpulan Data

Pengumpulan data adalah langkah awal yang sangat penting dalam proyek machine learning. Data yang tepat akan menentukan kualitas model yang kita bangun. Mari kita lihat lebih dalam tentang cara mengumpulkan data dan berbagai sumber yang bisa dimanfaatkan.

Sumber Data

Data bisa kamu ambil dari berbagai sumber yang memiliki keunikan. Berikut beberapa di antaranya.



- **Database Internal**
  Jika kamu bekerja di perusahaan, coba cek database internal yang

berisi informasi penting seperti data pelanggan atau transaksi. Data ini bisa kamu ambil langsung dari *database management system* (DBMS) seperti **MySQL** atau **PostgreSQL**. Kamu bisa menggunakan query SQL untuk mendapatkan data yang sesuai dengan kebutuhanmu.

- **Open Source Data Platforms**
  Banyak platform seperti Kaggle menyediakan dataset yang siap digunakan untuk analisis dan proyek machine learning. Pada Kaggle, kamu bisa menemukan beragam data dari berbagai bidang—mulai dari kesehatan, cuaca, hingga keuangan. Jangan lupa untuk menjelajahi kompetisi yang ada di sana, kamu bisa belajar banyak dari proyek-proyek yang telah dikerjakan orang lain, *lo!*
- **API (Application Programming Interface)**
  Banyak platform yang menyediakan API untuk akses data yang lebih mudah. Misalnya, jika ingin menganalisis sentimen di Twitter, kamu bisa memanfaatkan Twitter API. Untuk ini, kamu perlu sedikit pemahaman tentang pemrograman dan cara autentikasi menggunakan kunci API. Tertarik untuk mencoba?
- **Media Sosial**
  Platform seperti Instagram dan Facebook merupakan tambang emas untuk opini publik. Dengan menggunakan scraping tools atau API, kamu bisa mengumpulkan komentar dan interaksi pengguna.
- **Situs Web dan Forum Diskusi**
  Data dari situs review seperti Yelp atau forum seperti Reddit juga bisa sangat berharga. Dengan teknik web scraping, kamu bisa mendapatkan berbagai opini pengguna tentang produk dan layanan. Jangan lupa untuk selalu cek kebijakan situs tersebut ya!
- **Survei dan Kuesioner**
  Jika kamu merasa data yang ada tidak cukup, mengadakan survei atau kuesioner bisa jadi solusi. Alat seperti Google Forms atau SurveyMonkey memudahkan kamu untuk membuat dan mendistribusikan kuesioner. Sudahkah kamu pernah mencoba?

Cara Mengambil Data

Setelah menemukan sumber data, langkah berikutnya adalah mengambil data tersebut. Berikut adalah beberapa metode yang bisa kamu coba.

- **Penggunaan Skrip Otomatis**
  Jika data tersedia di situs web atau API, kamu bisa membuat *script* menggunakan Python. Misalnya, gunakan library requests untuk memanggil API dan pandas untuk menyimpan data dalam format yang terstruktur. Mau belajar cara bikin skripnya?
- **Web Scraping**
  Jika situs tidak punya API, kamu bisa menggunakan web scraping. Pastikan untuk memeriksa izin scraping dari situs tersebut. Dengan library seperti BeautifulSoup, kamu bisa dengan mudah mengambil informasi dari halaman web.
- **Data Extraction Tools**
  Jika tidak mau repot dengan pemrograman, ada alat seperti Octoparse atau ParseHub yang bisa membantumu mengambil data. Alat ini cocok untuk kamu yang lebih suka antarmuka visual.

Memastikan Kualitas Data

Mengumpulkan data saja tidak cukup; kamu juga harus memastikan data berkualitas. Berikut beberapa tips untuk menjamin kualitas data.

- **Representatif:** Pastikan data mewakili populasi yang ingin kamu analisis. Misalnya, jika analisis untuk remaja, kumpulkan data dari mereka yang berusia remaja saja.
- **Bersih dari Duplikasi:** Cek dan hapus data duplikat untuk menghindari bias. Coba gunakan fungsi di Pandas untuk memeriksa ini!
- **Konsistensi:** Pastikan format data konsisten. Misalnya, jika mengumpulkan data tanggal, semua harus dalam format yang sama (seperti YYYY-MM-DD).
- **Pengujian Kualitas:** Periksa kualitas data dengan visualisasi atau statistik deskriptif untuk menemukan outlier.
- **Data yang Cukup:** Pastikan jumlah data cukup untuk model yang andal. Jika terbatas, coba teknik augmentasi data atau cari lebih banyak sumber.

Dengan pendekatan yang tepat dalam pengumpulan data, kamu bisa memastikan dasar proyek machine learning kamu kuat dan siap melangkah ke tahap berikutnya.

2. Exploratory Data Analysis

Setelah pengumpulan data, tahap berikutnya yang tidak kalah penting adalah **Exploratory Data Analysis (EDA)**. EDA adalah proses mengeksplorasi data yang sudah dibersihkan untuk mendapatkan wawasan dan menjawab pertanyaan analisis. Dalam proses ini, kamu akan menggunakan berbagai teknik statistik deskriptif untuk menemukan pola dan hubungan dalam data. Selain itu, visualisasi data juga sering digunakan untuk membantu memahami pola-pola yang ada.

Banyak praktisi data menganggap EDA sebagai tahap yang menarik dalam analisis data. Di sini, kamu bisa bereksperimen dengan data untuk menemukan informasi berharga, mengidentifikasi anomali, dan merumuskan kesimpulan dari hasil analisis.

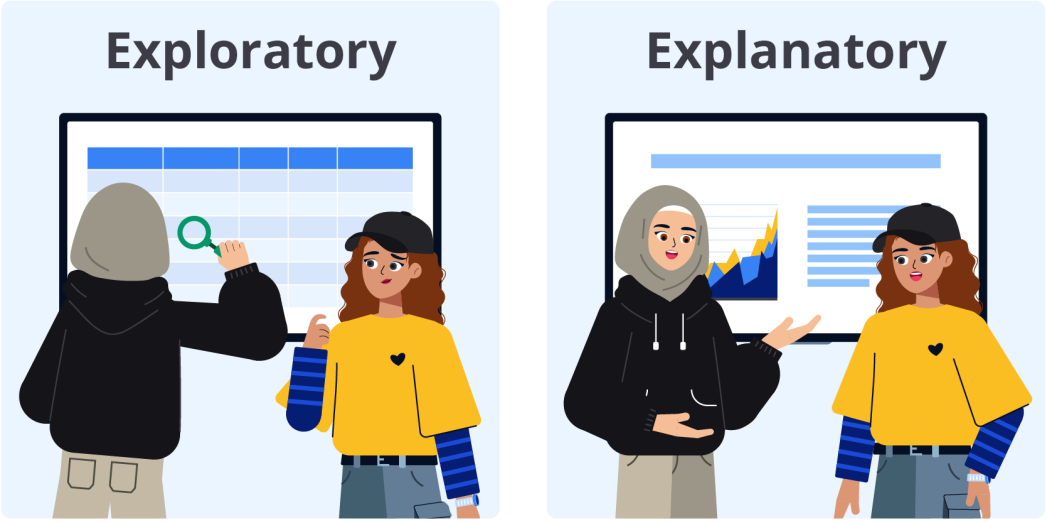
Sebagai calon *engineer*, kamu pasti akan menikmati tahap EDA ini, karena sangat penting dan tidak bisa dilewatkan. Sebelum itu, mari kita pahami perbedaan antara **Exploratory Data Analysis (EDA)** dan **Explanatory Data Analysis (ExDA)**.

Perbedaan EDA dan ExDA

Exploratory Data Analysis (EDA) adalah proses untuk memahami struktur dan pola dalam data. Tujuan utamanya adalah menemukan

wawasan tersembunyi serta hubungan antar variabel tanpa perlu memiliki hipotesis awal.

Dalam EDA, kamu akan menggunakan teknik statistik deskriptif, seperti menghitung mean dan median, serta membuat visualisasi data, seperti histogram dan box plot. Proses ini bersifat fleksibel dan iteratif sehingga kamu bisa mencoba berbagai pendekatan untuk mendapatkan wawasan yang lebih dalam. Biasanya, EDA dilakukan oleh analis data atau data scientist yang berinteraksi langsung dengan data.



Sementara itu, **Explanatory Data Analysis (ExDA)** bertujuan untuk mengkomunikasikan temuan dari analisis data kepada audiens yang lebih luas, seperti manajemen atau klien. Di sini, fokusnya adalah menyampaikan informasi dengan jelas dan meyakinkan.

ExDA menggunakan visualisasi yang terarah dan narasi yang kuat untuk mendukung kesimpulan yang dihasilkan. Metodologi dalam ExDA

terstruktur, dimulai dari pernyataan masalah hingga mencapai kesimpulan yang jelas. Audiens dari ExDA biasanya adalah *stakeholder* dan pengambil keputusan yang memerlukan informasi untuk membantu mereka membuat keputusan bisnis yang tepat.

Misalnya, dalam proyek yang bertujuan menemukan faktor-faktor yang memengaruhi penjualan, EDA digunakan untuk menganalisis hubungan antara variabel, seperti harga dan waktu promosi. Selama proses ini, kamu dapat menemukan pola tak terduga atau anomali dalam data yang mungkin tidak terlihat sebelumnya. Hasil dari EDA biasanya berupa wawasan baru, hipotesis yang dapat diuji, dan pemahaman yang lebih mendalam tentang data yang dianalisis.

Setelah menemukan temuan penting melalui EDA, tahap selanjutnya adalah ExDA. Di sini, kamu akan membuat laporan yang menjelaskan faktor-faktor yang memengaruhi penjualan kepada manajemen. Laporan ini akan dilengkapi dengan grafik sederhana dan narasi yang jelas, sehingga manajemen dapat memahami informasi dengan mudah dan mengambil keputusan yang tepat berdasarkan hasil analisis. Hasil dari ExDA umumnya berupa laporan akhir, presentasi, atau dashboard yang menyampaikan hasil analisis dengan cara yang informatif dan mudah dipahami oleh audiens.

3. Data Preprocessing atau Pemrosesan Data

Data preprocessing dalam machine learning bisa diibaratkan seperti persiapan bahan-bahan sebelum memasak. Ketika kamu ingin memasak

suatu hidangan, kamu tidak bisa langsung memasukkan semua bahan mentah ke dalam panci. Kamu perlu mencuci, memotong, dan mengolah bahan-bahan tersebut agar menjadi siap saji.

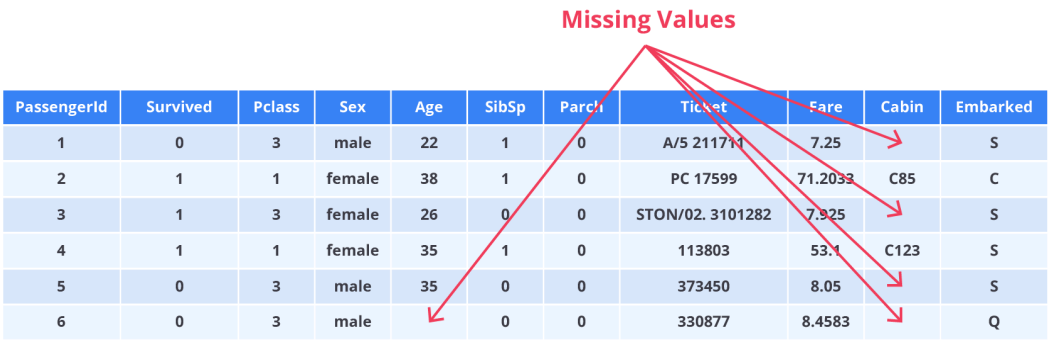
Begitu juga dalam data preprocessing, kita perlu membersihkan dan mempersiapkan data mentah—seperti mengatasi data yang hilang, menangani data yang tidak wajar (*outliers*), dan mengubah format data—agar data tersebut siap untuk digunakan oleh model *machine learning*, sama seperti bahan masakan yang telah dipersiapkan siap untuk dimasak dan disajikan.

Proses ini melibatkan berbagai teknik dan transformasi untuk memastikan bahwa data yang digunakan memiliki kualitas tinggi, konsisten, dan sesuai dengan tujuan analisis atau pemodelan. Dengan kata lain, langkah ini mengubah dan memodifikasi fitur-fitur data menjadi bentuk yang lebih mudah dipahami dan diproses oleh algoritma machine learning.

Berikut adalah langkah-langkah yang biasanya dilakukan dalam data preprocessing.

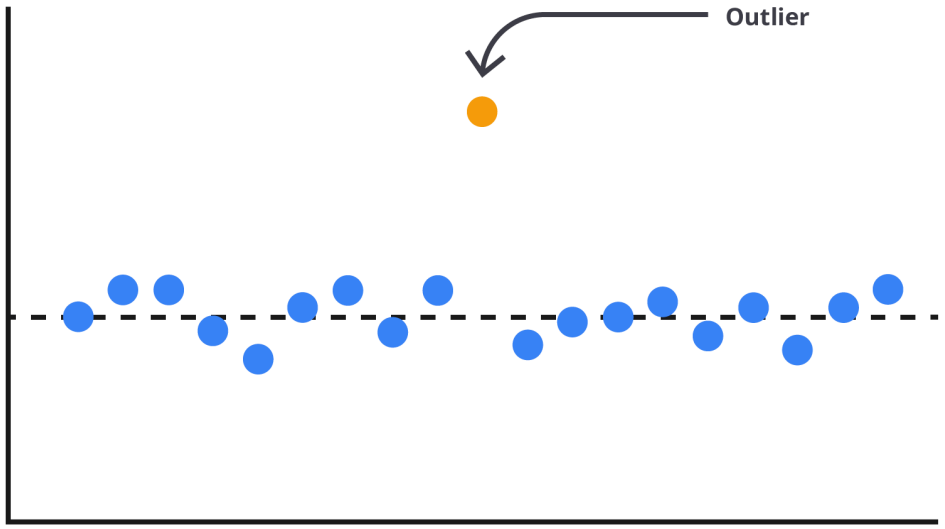
- **Menangani Data yang Hilang**
  Data yang hilang atau **missing values** adalah masalah umum dalam dataset. Kamu bisa mengatasi masalah ini dengan cara menghapus data yang hilang atau menggunakan **imputasi**, yaitu mengganti nilai yang hilang dengan nilai lain, seperti mean/rata-rata, median, atau

modus.



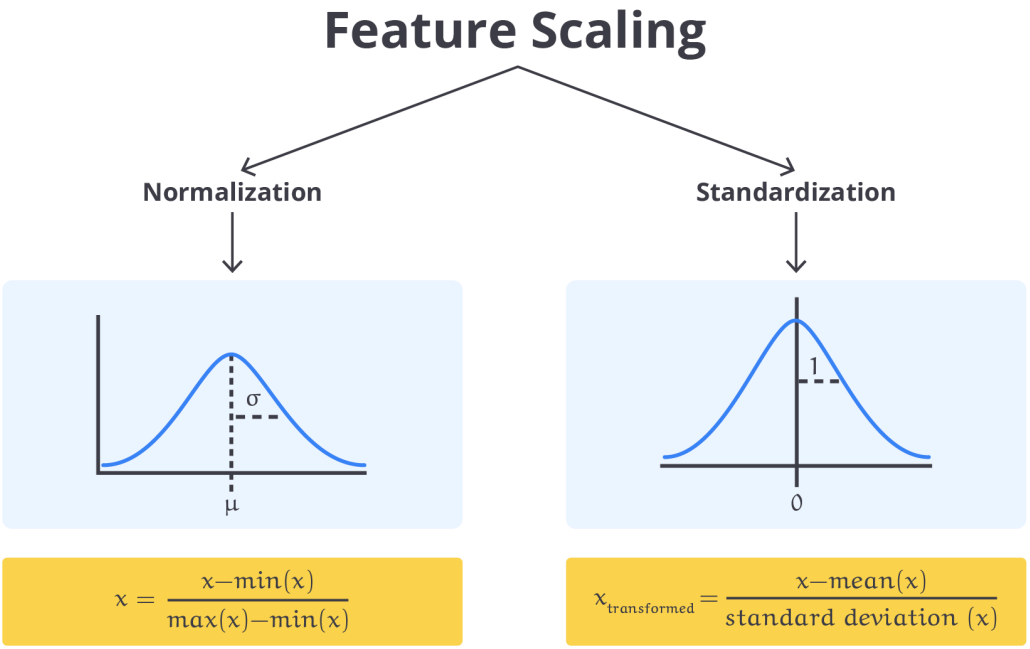
PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	male	22	1	0	A/5 211711	7.25		S
2	1	1	female	38	1	0	PC 17599	71.2033	C85	C
3	1	3	female	26	0	0	STON/02. 3101282	7.925		S
4	1	1	female	35	1	0	113803	53.1	C123	S
5	0	3	male	35	0	0	373450	8.05		S
6	0	3	male		0	0	330877	8.4583		Q

- **Mengatasi *Outliers***
  Outliers adalah nilai yang jauh berbeda dari mayoritas data lainnya dan bisa memengaruhi kinerja model. Beberapa cara untuk menangani *outliers* adalah dengan menghapus, melakukan transformasi data, atau mengubah nilai agar lebih mendekati distribusi normal.



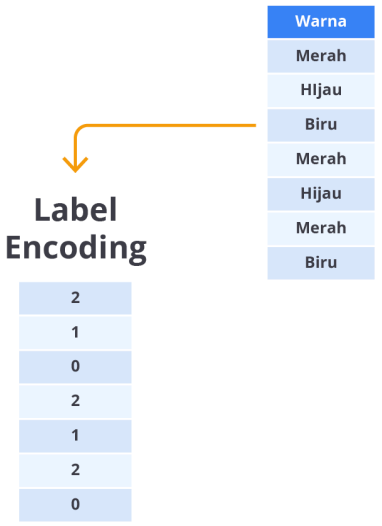
- **Transformasi Data**
  Langkah ini bertujuan untuk memastikan data berada dalam format yang sesuai untuk model. Teknik-teknik transformasi yang umum

digunakan meliputi:

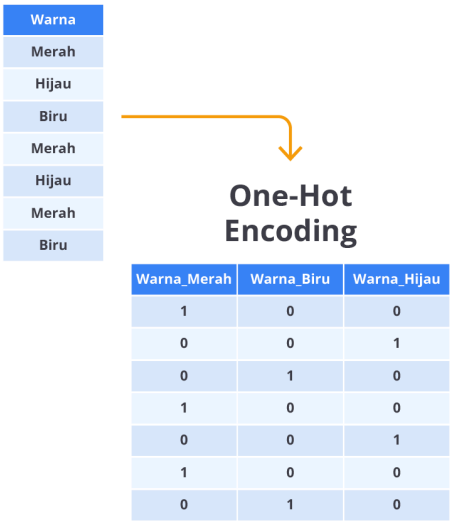


  - **Normalisasi**: Mengubah skala data sehingga berada dalam rentang tertentu, biasanya antara 0 dan 1.
  - **Standardisasi**: Mengubah data sehingga memiliki distribusi dengan rata-rata 0 dan standar deviasi 1, sering digunakan untuk data yang terdistribusi Gaussian.
- **Encoding Variabel Kategorikal**
  Karena model machine learning hanya dapat bekerja dengan data numerik, variabel kategorikal perlu diubah menjadi bentuk numerik. Beberapa teknik yang digunakan adalah:

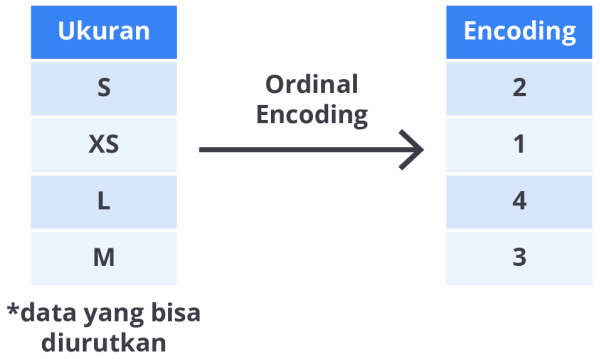
- **Label Encoding**: mengubah kategori menjadi label numerik, cocok untuk variabel ordinal.



- **One-Hot Encoding**: mengubah setiap kategori menjadi kolom biner terpisah (0 atau 1) untuk kategori yang tidak memiliki hubungan ordinal.



- **Ordinal Encoding**: mengubah kategori menjadi label numerik berdasarkan urutan atau tingkatan.



Dengan menjalankan langkah-langkah di atas, kamu akan memastikan bahwa data yang digunakan untuk model machine learning berkualitas tinggi dan siap untuk dianalisis.

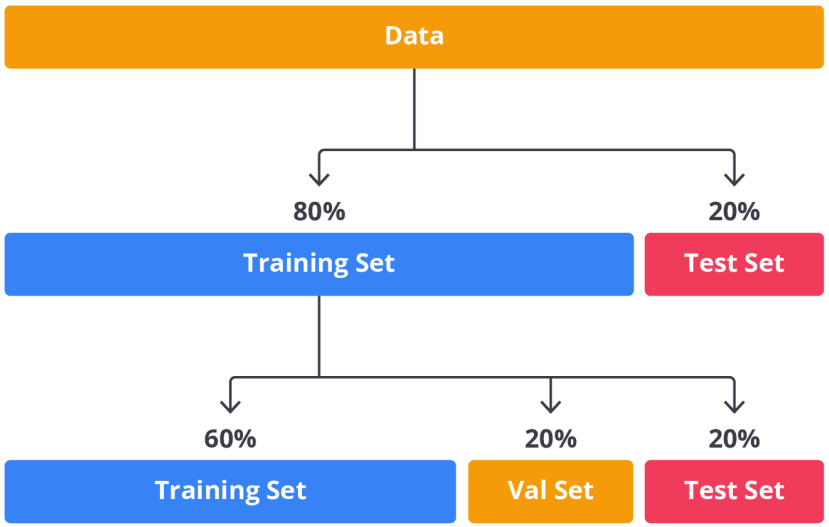
4. Model Training atau Pelatihan Model

Pembangunan model atau **model development** adalah tahap membuat, melatih, dan menguji model machine learning untuk menghasilkan prediksi yang akurat. Tahap ini sangat penting karena kualitas model yang dibangun akan mempengaruhi seberapa baik model tersebut dapat bekerja di dunia nyata. Berikut ini penjelasan lebih rinci mengenai proses pembangunan model.

Pembagian Data (Train-Test Split)

Sebelum melatih model, sangat penting untuk membagi dataset menjadi beberapa bagian agar kita bisa mengevaluasi kinerja model secara objektif. Tujuan utamanya adalah memastikan bahwa model tidak hanya bekerja dengan baik pada data yang dikenal (data latih), tetapi juga mampu memprediksi dengan baik pada data baru.

Pembagian ini biasanya dilakukan sebagai berikut.



- **Training Set (Data Latih)**
  Bagian terbesar dari dataset digunakan untuk melatih model dan menemukan pola. Pada tahap ini, model "belajar" dari data dan mencari hubungan antara variabel input dan output. Biasanya, 70-80% dari dataset dialokasikan untuk data latih.
- **Test Set (Data Uji)**
  Sekitar 20-30% dari dataset dialokasikan sebagai data uji. Setelah model selesai dilatih, data uji digunakan untuk mengevaluasi apakah model dapat memberikan prediksi yang akurat pada data baru yang tidak pernah dilihat sebelumnya. Evaluasi ini membantu mengetahui seberapa baik model akan bekerja di dunia nyata.
- **Validation Set (Opsional)**
  Validation set digunakan saat kita ingin melakukan **hyperparameter**

tuning (penyesuaian parameter model). Dengan validation set, kamu bisa menguji perubahan parameter tanpa menggunakan data uji, sehingga data uji tetap “murni” untuk evaluasi akhir. Biasanya, validation set diambil dari data latih, misalnya 10-20% dari data latih.

Berikut adalah contoh skema pembagian data untuk total 1200 data.

Jika **tanpa** menggunakan data validasi.

- **900 data** → Data latih (training set)
- **300 data** → Data uji (test set)

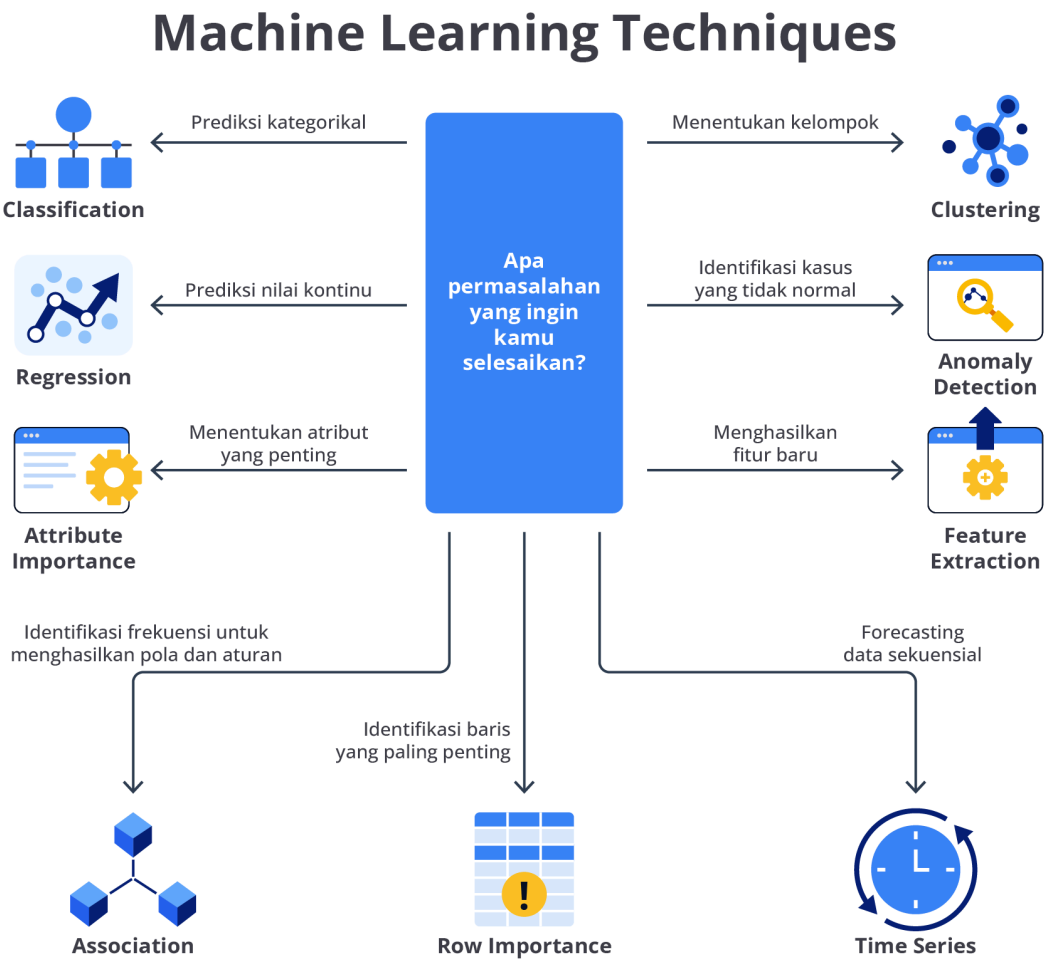
Jika **menggunakan** data validasi,

- **700 data** → Data latih (training set)
- **200 data** → Data validasi (jika diperlukan)
- **300 data** → Data uji (test set)

Memilih Jenis Algoritma Supervised Learning dan Unsupervised Learning

Pada tahap ini, kamu memilih jenis **algoritma machine learning** berdasarkan tipe masalah dan data yang kamu miliki. Ada dua pendekatan

utama, yaitu supervised learning dan unsupervised learning.



Supervised Learning (Pembelajaran Terawasi)

Supervised learning digunakan ketika dataset memiliki **label** atau **target** yang jelas. Artinya, setiap data latih sudah berpasangan antara fitur input (misalnya luas rumah) dan output atau label (misalnya harga rumah). Model ini belajar dengan memetakan hubungan antara input dan output, sehingga dapat melakukan prediksi pada data baru dengan pola yang serupa.

Contoh Kasus Supervised Learning

1. **Klasifikasi:** Klasifikasi adalah metode dalam machine learning yang digunakan untuk mengelompokkan data ke dalam **kategori atau label** tertentu. Dalam klasifikasi, model dilatih untuk memprediksi kelas dari data input berdasarkan fitur yang ada, dengan tujuan untuk mengidentifikasi kategori mana yang paling sesuai. Misalnya, model dibuat untuk menentukan apakah sebuah email adalah **spam** atau **bukan spam** berdasarkan fitur-fitur seperti kata-kata tertentu, pengirim, atau tautan dalam email.
2. **Regresi:** Regresi adalah metode dalam machine learning yang digunakan untuk memprediksi **nilai kontinu** berdasarkan variabel input. Dalam regresi, model dilatih untuk memperkirakan hubungan antara fitur dan nilai target, sehingga dapat menghasilkan prediksi numerik. Misalnya, model dapat digunakan untuk memprediksi **harga rumah** berdasarkan fitur seperti luas rumah, jumlah kamar, dan lokasi. Di sini, harga rumah adalah variabel kontinu yang menjadi target.

Algoritma Supervised Learning yang Populer

- **Klasifikasi**
  - **K-Nearest Neighbors (KNN):** membandingkan data baru dengan data terdekat dalam dataset.
  - **Decision Tree:** membuat pohon keputusan berdasarkan fitur untuk memprediksi kategori.
  - **Random Forest:** kombinasi beberapa pohon keputusan untuk mendapatkan prediksi yang lebih akurat.
- **Regresi**
  - **Linear Regression:** mencari garis lurus terbaik untuk memprediksi nilai kontinu.
  - **Ridge Regression:** versi regresi linear dengan regularisasi untuk menghindari overfitting.

Unsupervised Learning (Pembelajaran Tak Terawasi)

Unsupervised learning diterapkan pada dataset **tanpa label atau target**. Tujuannya adalah untuk menemukan pola tersembunyi, kelompok, atau struktur dalam data. Model ini berusaha membuat interpretasi sendiri dari data tanpa informasi spesifik mengenai kategori atau nilai output.

Contoh Kasus Unsupervised Learning

1. **Clustering (Pengelompokan)**
   Digunakan untuk membagi data ke dalam beberapa kelompok atau segmen. Misalnya, perusahaan dapat mengelompokkan pelanggan berdasarkan pola belanja mereka agar bisa membuat strategi pemasaran yang lebih tepat.
2. **Dimensionality Reduction (Pengurangan Dimensi)**
   Digunakan untuk mengurangi jumlah fitur dalam dataset yang sangat besar sambil tetap mempertahankan informasi penting. Contohnya, teknik **PCA (Principal Component Analysis)** mengubah fitur-fitur yang ada menjadi komponen baru agar analisis lebih efisien.

Algoritma Unsupervised Learning yang Populer

- **K-Means**
  Algoritma ini mencari titik pusat (centroid) dan mengelompokkan data ke dalam sejumlah cluster berdasarkan jarak terdekat dengan centroid tersebut.
- **DBSCAN**
  Algoritma ini mengelompokkan data berdasarkan kepadatan titik data. Cocok digunakan saat dataset memiliki **outlier** karena outlier tidak akan dimasukkan ke dalam cluster mana pun.
- **Hierarchical Clustering**
  Algoritma ini membentuk struktur **hierarki** dengan membuat cluster berulang kali dari data yang paling mirip, hingga seluruh data tergabung dalam satu cluster besar.

Pelatihan Model (Training)

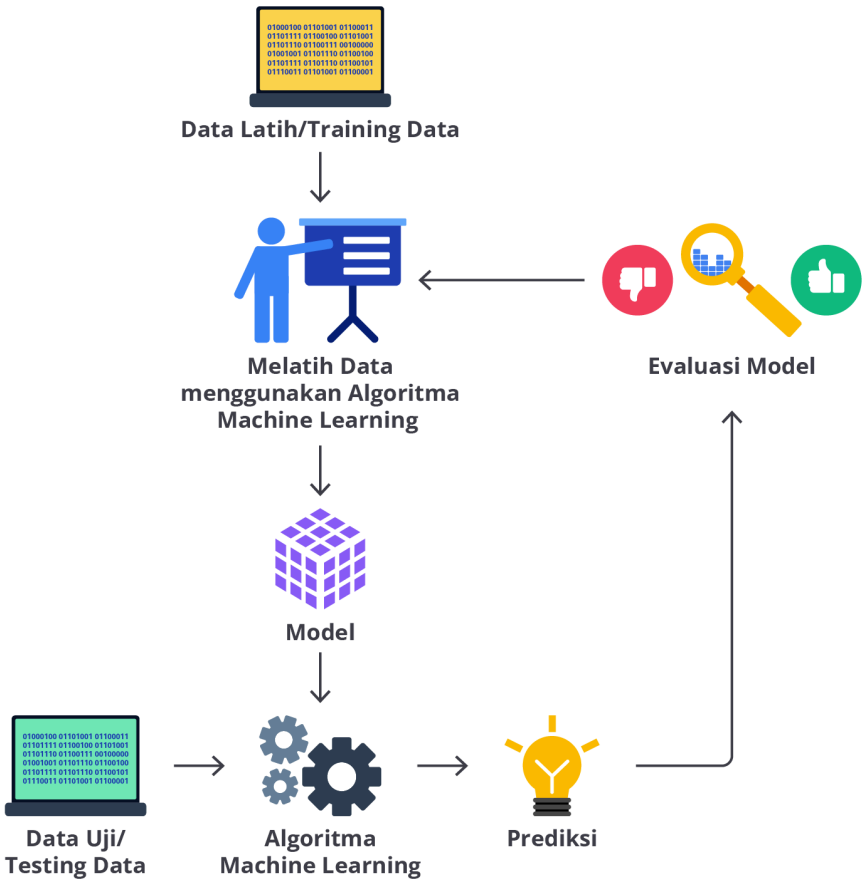
Pada tahap ini, model mulai “belajar” dari data latih. Proses pelatihan ini berbeda untuk **supervised** dan **unsupervised learning**. Namun, intinya model berusaha menemukan pola atau fungsi terbaik yang bisa menghubungkan fitur dengan output (untuk supervised) atau menemukan struktur tersembunyi (untuk unsupervised).

5. Model Evaluation atau Evaluasi Model

Sekarang kita sudah sampai di tahap **evaluasi model**. Pada bagian ini, kita akan melakukan pengecekan pada model machine learning yang kita buat apakah **sudah bekerja dengan baik** atau belum.

Bayangkan kamu sedang mempersiapkan kue untuk teman-teman—meskipun kamu sudah mengikuti resep dengan benar, kamu tetap harus mencicipi hasil akhirnya, bukan? Sama halnya dengan model machine learning. Kita perlu memastikan bahwa model tidak hanya “kelihatan bagus”, tapi juga benar-benar memberikan hasil yang akurat dan

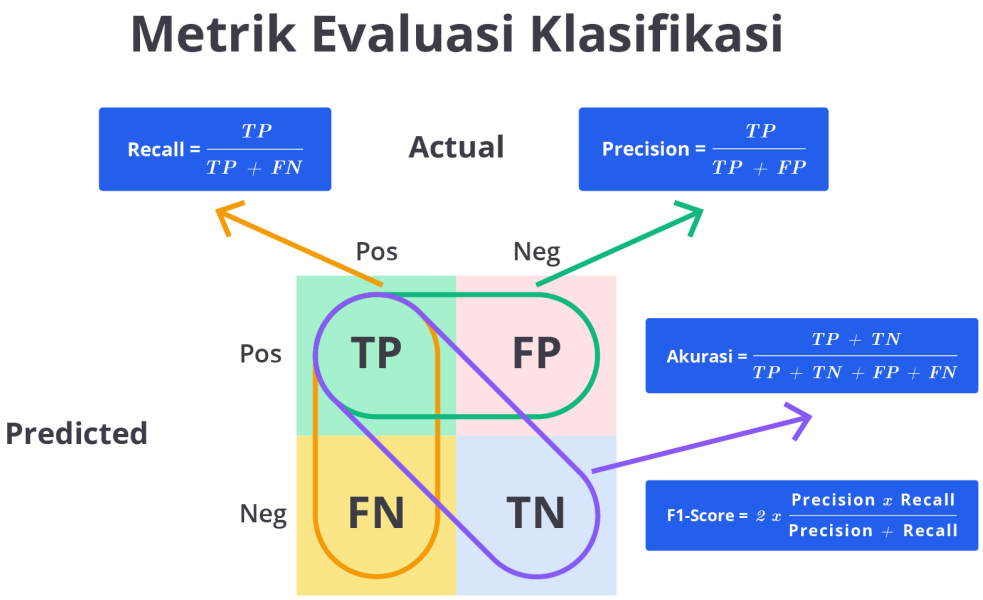
konsisten.



Proses ini penting untuk memastikan bahwa model bisa bekerja baik tidak hanya pada data yang sudah dilatih, tapi juga pada **data baru**. Model yang terlihat hebat pada data latih belum tentu akan bekerja sama baiknya di dunia nyata karena mungkin model tersebut terlalu hafal dengan data yang ada (alias **overfitting**).

Evaluasi Model: Supervised vs. Unsupervised Learning

1. **Evaluasi Supervised Learning**
   Pada supervised learning, kita mengukur seberapa akurat model memprediksi **label** yang benar dengan data uji. Beberapa metrik yang sering digunakan.



- **Akurasi:** persentase prediksi yang benar.
- **Precision & Recall:** precision mengukur akurasi hasil positif, sedangkan Recall melihat seberapa banyak kasus positif terdeteksi.
- **F1-Score:** rata-rata harmonis dari precision dan recall untuk keseimbangan keduanya.
- **Confusion Matrix:** menunjukkan jumlah prediksi benar dan salah dalam bentuk tabel (True/False Positive dan Negative).

✏️ *Contoh:* memprediksi apakah email termasuk spam atau bukan untuk kasus klasifikasi dan memprediksi perbedaan nilai untuk kasus regresi.

2. **Evaluasi Unsupervised Learning**
   Karena tidak ada label, evaluasi di unsupervised learning lebih fokus pada **struktur dan pola** dalam data. Beberapa metrik umum:
- **Silhouette Score:** mengukur seberapa baik objek dalam satu cluster mirip satu sama lain.

- **Inertia:** total jarak antara titik data dan pusat cluster, semakin kecil semakin baik.
- **Davies-Bouldin Index:** semakin kecil nilainya, semakin baik kualitas clustering.

✏️ *Contoh:* mengelompokkan pelanggan berdasarkan kebiasaan belanja.

Dengan evaluasi yang tepat, kamu bisa memastikan model bekerja optimal, baik untuk memprediksi (supervised) maupun menemukan pola (unsupervised).

6. Deployment

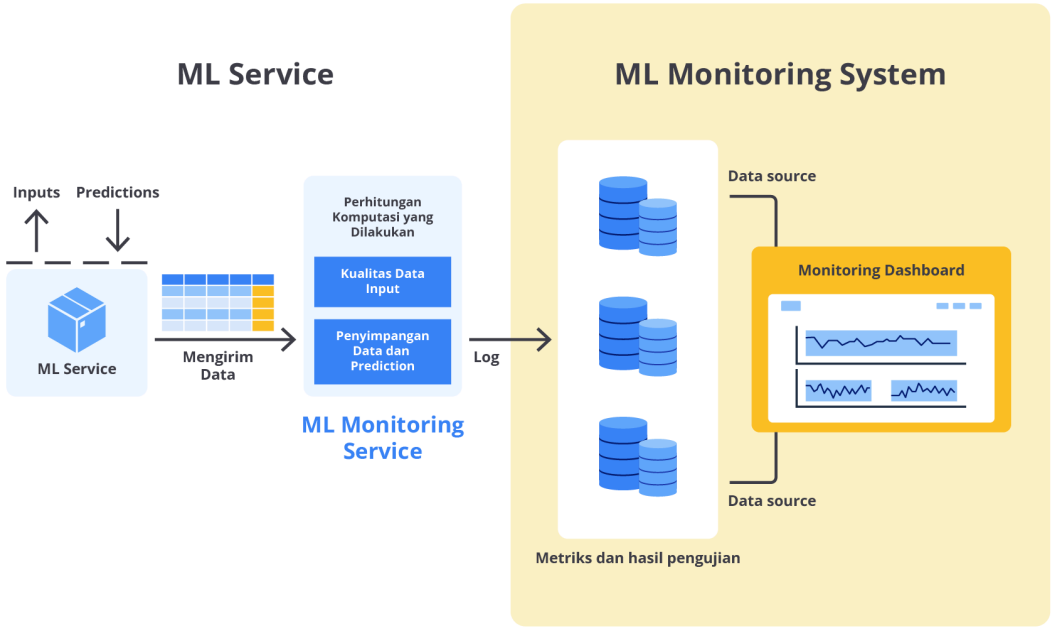
Deployment atau penerapan model adalah tahapan ketika model machine learning yang sudah dilatih dan diuji diterapkan ke dunia nyata agar bisa digunakan. Model ini bisa diintegrasikan ke dalam **aplikasi web, mobile, API, atau perangkat IoT.** Setelah diterapkan, tentunya penting untuk memantau performanya secara rutin karena data bisa berubah seiring waktu.

Jika akurasi model menurun, perlu dilakukan **retraining** dengan data baru. Selain itu, optimasi seperti kompresi model dan *load balancing* diperlukan agar model bisa berjalan cepat dan efisien, terutama saat menangani banyak pengguna. Tidak kalah penting lagi, aspek keamanan dan privasi

data juga harus diperhatikan agar penerapan model tetap aman dan sesuai regulasi.

7. Monitoring atau Pemantauan

Monitoring atau pemantauan adalah tahapan terakhir yang sangat penting setelah penerapan model machine learning yang mengharuskanmu aktif mengawasi performa model dalam lingkungan nyata. Tahap ini bertujuan untuk memastikan model berfungsi dengan baik dan memberikan hasil yang akurat seiring waktu.



Beberapa hal yang perlu diperhatikan dalam proses monitoring antara lain sebagai berikut.

- **Kinerja Model**

Secara berkala, evaluasi metrik kinerja model seperti akurasi, presisi, recall, atau F1-score, tergantung pada jenis masalah yang dipecahkan. Hal ini membantu mendeteksi jika model mulai kehilangan akurasi.

- **Data Drift**

Perhatikan perubahan dalam data input yang mungkin terjadi seiring waktu, yang dikenal sebagai *data drift*. *Data drift* dapat mempengaruhi kemampuan model untuk membuat prediksi yang tepat. Misalnya, jika kebiasaan konsumen berubah, model yang sebelumnya akurat mungkin mulai menghasilkan hasil yang tidak relevan.

- **Feedback Loop**

Kumpulkan umpan balik dari pengguna dan sistem untuk terus meningkatkan model. Umpan balik ini bisa membantu dalam memperbaiki fitur atau bahkan pengembangan model lebih lanjut.

- **Maintenance**

Siapkan rencana pemeliharaan untuk melakukan retraining model dengan data baru agar tetap relevan dan akurat. Proses ini bisa melibatkan pembaruan data, fine-tuning parameter, atau penggantian model sepenuhnya jika diperlukan.

Dengan monitoring yang tepat, kamu bisa memastikan bahwa model machine learning tetap efektif, responsif terhadap perubahan, dan dapat terus memberikan nilai tambah dalam pengambilan keputusan.