# MAKALAH

# PRA PEMROSESAN DATA II

# Feature Engineering - Feature Selection dan Reduksi Dimensi (PCA)


## Disusun Oleh:


1. **Agung Adi Rangga** **(105841102323)**
2. **Suriani** **(105841117223)**
3. **Nurul Afsari** **(105841117823)**
4. **Karnis** **(105841102123)**
5. **Muhammad Ilham** **(105841101823)**
6. **Rama Bramanthiyo Susanto Putra** **(105841115123)**
7. **Reno** **(105841103423)**


**Dosen Pembimbing:**

**RUNAL REZKIAWAN, S.Kom.,M.T**


**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH MAKASSAR**


**2025**

# KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan makalah yang berjudul "Pra Pemrosesan Data II: Feature Engineering - Feature Selection dan Reduksi Dimensi (PCA)" ini tepat pada waktunya.

Makalah ini disusun sebagai respons terhadap kebutuhan yang semakin mendesak dalam bidang *Machine Learning* dan *Data Science*, di mana efisiensi dan akurasi model sangat bergantung pada kualitas dan representasi data input. Dalam ekosistem data modern, menghadapi *dataset* dengan dimensi tinggi (banyak fitur) yang mengandung kompleksitas, redudansi, dan potensi *overfitting* adalah tantangan yang jamak.

Oleh karena itu, makalah ini berfokus pada pembahasan mendalam mengenai tiga pilar penting dalam pra-pemrosesan data tingkat lanjut: Feature Engineering, yang berfungsi meningkatkan kualitas fitur; Feature Selection, yang memilih subset fitur paling relevan; dan Principal Component Analysis (PCA), sebagai teknik reduksi dimensi yang efisien.

Penulis berharap makalah ini dapat memberikan kontribusi signifikan bagi pembaca, khususnya mahasiswa, praktisi, dan peneliti yang tertarik untuk mengoptimalkan alur kerja *Machine Learning* melalui pemahaman yang komprehensif tentang teknik-teknik seleksi dan reduksi fitur.

Penulis menyadari bahwa makalah ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun dari berbagai pihak sangat diharapkan untuk penyempurnaan di masa mendatang.

Makassar, 29 Oktober 2025

Penulis

# DAFTAR ISI

# BAB I

# PENDAHULUAN

## 1.1. Latar Belakang

Dalam konteks *Machine Learning* (ML) dan analisis data, kualitas dan dimensi *dataset* secara langsung memengaruhi kinerja, efisiensi, dan interpretasi model. Proses pra-pemrosesan data merupakan tahap krusial, di mana Feature Engineering (Rekayasa Fitur) memegang peran vital dalam mengubah data mentah menjadi representasi fitur yang lebih bermakna dan efektif. Rekayasa Fitur adalah proses penciptaan fitur-fitur baru atau modifikasi fitur yang sudah ada dari data mentah untuk meningkatkan kekuatan prediktif model. Teknik ini meliputi penanganan berbagai tipe data seperti numerik, kategorikal, teks, deret waktu, citra, hingga data geospasial. Melalui praktik terbaik Feature Engineering, seperti menghindari *data leakage* dan memastikan relevansi fitur, kualitas *dataset* secara fundamental ditingkatkan. Meskipun *Feature Engineering* bertujuan untuk menciptakan fitur yang superior, seringkali *dataset* yang dihasilkan memiliki dimensi yang sangat tinggi. *Dataset* berdimensi tinggi ini menimbulkan beberapa masalah:

1. Overfitting: Model terlalu spesifik pada data pelatihan dan gagal dalam data yang tidak terlihat.
2. Efisiensi Komputasi: Meningkatkan waktu pelatihan secara signifikan.
3. Interpretasi: Membuat model menjadi *black-box* yang sulit dipahami.

Untuk mengatasi tantangan *curse of dimensionality* ini, muncul dua pendekatan utama dalam mereduksi fitur:

**a. Feature Selection (Seleksi Fitur):**

Proses memilih subset fitur asli yang paling relevan dan penting, sambil meninggalkan fitur yang kurang informatif atau berlebihan. Pendekatan ini mempertahankan fitur aslinya, sehingga hasil model tetap mudah diinterpretasikan. Metode utama dalam seleksi fitur dibagi menjadi tiga kategori, yaitu Filter, Wrapper, dan Embedded.

**b. Dimensionality Reduction (Reduksi Dimensi):**

Teknik yang bertujuan mengompresi data dengan memproyeksikan fitur ke ruang berdimensi lebih rendah. Berbeda dengan seleksi fitur, reduksi dimensi tidak hanya memilih fitur tetapi juga menciptakan komponen-komponen baru. Salah satu metode paling populer dan linier adalah Principal Component Analysis (PCA). PCA bekerja dengan mengidentifikasi arah (komponen utama) di mana variasi data paling maksimal, sehingga dapat mengurangi dimensi secara efisien. Meskipun PCA unggul dalam kompresi dan efisiensi waktu eksekusi , interpretasi hasil reduksi dimensinya dapat menjadi lebih kompleks.

Makalah ini disusun untuk mendalami ketiga konsep ini *Feature Engineering*, *Feature Selection*, dan *Reduksi Dimensi* (PCA) dengan fokus komparatif pada bagaimana kedua teknik reduksi fitur (Seleksi dan PCA) dapat diterapkan untuk mengoptimalkan alur kerja *Machine Learning*.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, permasalahan utama yang akan dibahas dalam makalah ini dapat dirumuskan sebagai berikut:

1. Apa peran pra-pemrosesan data dalam meningkatkan kinerja model *machine learning*?
2. Bagaimana proses *feature engineering* dilakukan untuk menghasilkan fitur yang relevan dan berkualitas?
3. Bagaimana *feature selection* membantu memilih fitur yang paling penting bagi model?
4. Apa tujuan dan cara kerja *dimensionality reduction*, khususnya metode *Principal Component Analysis (PCA)*?
5. Bagaimana hubungan antara *feature selection* dan *dimensionality reduction* dalam mengoptimalkan model *machine learning*?

## 1.3. Tujuan Penelitian

Tujuan dari penulisan makalah ini adalah:

1. Untuk memahami konsep dan peran pra-pemrosesan data dalam meningkatkan akurasi serta efisiensi model *machine learning*.
2. Untuk menjelaskan proses dan manfaat *feature engineering* dalam menghasilkan fitur yang lebih bermakna dan relevan.
3. Mengidentifikasi cara kerja serta pentingnya *feature selection* dalam memilih fitur yang paling berpengaruh terhadap hasil prediksi.
4. Memahami konsep *dimensionality reduction* dan menjelaskan penerapan *Principal Component Analysis (PCA)* sebagai metode reduksi dimensi.
5. Menganalisis hubungan antara *feature selection* dan *dimensionality reduction* dalam upaya mengoptimalkan performa model *machine learning*.

# BAB II

# PEMBAHASAN

## 2.1. Konsep Pra-Pemrosesan Data dalam Machine Learning

Pra-pemrosesan data dalam machine learning adalah proses awal yang dilakukan sebelum data digunakan untuk melatih model. Tujuannya adalah mengubah data mentah menjadi bentuk yang bersih, terstruktur, dan siap diolah agar model dapat belajar dengan lebih akurat dan efisien. Data dari dunia nyata sering kali tidak sempurna, misalnya mengandung nilai yang hilang, duplikasi, kesalahan, atau memiliki format yang tidak seragam. Karena itu, tahapan ini menjadi sangat penting untuk memastikan kualitas data yang digunakan. Proses pra-pemrosesan biasanya dimulai dengan pembersihan data. Pada tahap ini, data diperiksa untuk mendeteksi dan memperbaiki masalah seperti nilai yang hilang, data yang salah, atau adanya outlier yang ekstrem. Nilai yang hilang bisa diisi dengan nilai rata-rata, median, atau modus, tergantung pada jenis datanya. Duplikasi dihapus, dan kesalahan input seperti ejaan atau format yang tidak konsisten disesuaikan agar data menjadi seragam.

Tahap berikutnya adalah transformasi data, yaitu mengubah format atau skala data agar sesuai dengan kebutuhan algoritma. Dalam banyak kasus, data numerik perlu dinormalisasi ke dalam rentang tertentu, misalnya antara 0 dan 1, atau distandarisasi agar memiliki rata-rata nol dan deviasi standar satu. Data kategorikal yang berbentuk teks juga diubah menjadi bentuk numerik menggunakan metode seperti label encoding atau one-hot encoding agar dapat diproses oleh model. Selain transformasi, dilakukan juga seleksi fitur, yaitu memilih variabel atau atribut yang paling relevan terhadap target prediksi. Langkah ini penting untuk menghindari kelebihan fitur yang justru dapat menurunkan kinerja model. Seleksi fitur bisa dilakukan dengan melihat korelasi antarvariabel, uji statistik, atau metode otomatis seperti *recursive feature elimination* dan *principal component analysis (PCA)*. Jika data berasal dari beberapa sumber, maka perlu dilakukan integrasi agar semua data dapat digabung menjadi satu kesatuan yang konsisten. Kadang juga dilakukan reduksi

data untuk mengurangi ukuran dataset tanpa mengurangi informasi penting di dalamnya, misalnya dengan teknik PCA atau sampling. Tahap akhir dari pra-pemrosesan adalah pembagian data menjadi dua atau tiga bagian, yaitu data pelatihan (training), data pengujian (testing), dan kadang juga data validasi. Pembagian ini bertujuan agar model dapat dilatih dan diuji pada data yang berbeda, sehingga kinerjanya bisa dievaluasi secara objektif dan tidak bias. Secara keseluruhan, pra-pemrosesan data berperan besar dalam menentukan keberhasilan model machine learning. Model yang baik tidak hanya bergantung pada algoritmanya, tetapi juga pada kualitas data yang digunakan. Melalui tahapan pembersihan, transformasi, dan pemilihan fitur yang tepat, data menjadi lebih representatif dan siap menghasilkan prediksi yang akurat.

## 2.2. Feature Engineering

Feature engineering adalah proses mengubah data mentah menjadi fitur yang lebih relevan dan bermakna untuk meningkatkan performa model machine learning. Ini mencakup pembuatan fitur baru, transformasi, pemilihan fitur yang paling berpengaruh, hingga pengurangan dimensi agar model dapat lebih memahami pola dalam data. Dengan teknik yang tepat, feature engineering dapat membantu model menghasilkan prediksi yang lebih akurat tanpa harus mengganti algoritma yang digunakan.

*Feature engineering* bukan hanya sekadar menyiapkan data, tetapi juga mengoptimalkannya agar model *machine learning* dapat bekerja lebih efektif. Proses ini melibatkan berbagai langkah, mulai dari memahami data, mentransformasikannya, hingga memilih fitur yang paling berpengaruh. Berikut tahapan utama dalam *feature engineering*:

### a. Memahami Data

Sebelum melakukan *feature engineering*, kamu perlu memahami karakteristik data yang akan digunakan. Ini mencakup analisis pola, distribusi, serta kemungkinan adanya *noise* dalam data. Berikut penjelasannya:

- Eksplorasi Data (EDA – Exploratory Data Analysis): EDA bertujuan untuk memahami struktur data dengan visualisasi dan statistik

deskriptif. Teknik seperti histogram, *scatter plot,* dan *box plot* membantu dalam mengidentifikasi pola dan anomali.

- Identifikasi Fitur yang Relevan: Tidak semua fitur dalam dataset memiliki dampak signifikan terhadap model. Dengan memahami domain masalah, kamu bisa menentukan fitur mana yang memiliki korelasi tinggi dengan target prediksi.
- Deteksi Outlier dan Missing Values: Data sering kali mengandung *outlier* atau nilai yang hilang (*missing values*), yang dapat memengaruhi hasil model. Teknik seperti IQR (Interquartile Range) dan mean/mode imputation sering digunakan untuk menanganinya.

**b. Transformasi Data**

Agar data lebih mudah dipahami oleh model, diperlukan beberapa teknik transformasi untuk menyesuaikan skala dan formatnya. Berikut ini prosesnya:

- Normalisasi dan Standarisasi: Normalisasi digunakan untuk mengubah skala data agar berada dalam rentang tertentu (misalnya 0 hingga 1), sedangkan standardisasi memastikan distribusi data memiliki rata-rata nol dan standar deviasi satu.
- Encoding Variabel Kategorikal: Data kategorik perlu dikonversi ke format numerik agar bisa diproses oleh algoritma *machine learning*. Teknik seperti *one-hot encoding*, *label encoding*, dan *target encoding* sering digunakan.
- Handling Missing Values: *Missing values* dapat diatasi dengan berbagai teknik, seperti menggantinya dengan mean/median, menggunakan model prediktif, atau bahkan menghapus data jika jumlahnya terlalu banyak.

**c. Feature Creation (Pembuatan Fitur Baru)**

Feature creation adalah proses menciptakan fitur baru dari fitur yang sudah ada untuk memperkaya informasi dalam data. Dengan pemahaman konteks yang baik, fitur baru dapat dibuat untuk menangkap hubungan

yang sebelumnya tidak terlihat. Contohnya, dari data "jumlah pembelian" dan "harga per unit" dapat dibuat fitur baru bernama "total pengeluaran". Dalam analisis pelanggan, selisih antara "tanggal pembelian terakhir" dan "tanggal pertama" dapat digunakan untuk membuat fitur "lama menjadi pelanggan". Fitur seperti ini sering kali meningkatkan performa model secara signifikan karena memberikan sudut pandang tambahan terhadap pola data. Membuat fitur baru dari data yang ada dapat membantu model memahami pola dengan lebih baik. Simak penjelasannya berikut ini:

- Kombinasi Fitur yang Ada: Menggabungkan dua atau lebih fitur yang saling berhubungan dapat menghasilkan informasi baru yang lebih bernilai, seperti menghitung rasio atau selisih antar fitur.
- Feature Extraction dari Teks, Gambar, atau Sinyal: Dalam NLP (Natural Language Processing), fitur dapat diekstrak menggunakan teknik seperti *TF-IDF* atau *word embeddings*. Untuk gambar, metode seperti *edge detection* atau *histogram of gradients* bisa diterapkan.
- Feature Selection Menggunakan Domain Knowledge: Memahami konteks data sangat penting dalam memilih fitur yang relevan. Pengetahuan domain membantu mengeliminasi fitur yang kurang berguna atau justru menambahkan fitur baru yang lebih informatif.

**d. Feature Selection (Pemilihan Fitur)**

Tidak semua fitur dalam dataset memiliki pengaruh terhadap target yang ingin diprediksi. Beberapa fitur bahkan dapat menyebabkan kebisingan (noise) dan menurunkan akurasi model. Oleh karena itu, feature selection dilakukan untuk memilih fitur yang paling relevan dan membuang yang tidak penting. Pemilihan fitur dapat dilakukan dengan analisis korelasi untuk melihat hubungan antarvariabel, uji statistik seperti chi-square dan ANOVA, atau metode berbasis model seperti feature importance pada Random Forest. Terdapat pula metode otomatis seperti Recursive Feature Elimination (RFE) dan Principal Component Analysis (PCA) yang

digunakan untuk mengurangi dimensi data tanpa kehilangan informasi penting. Setelah fitur dibuat, langkah selanjutnya memilih fitur yang paling berkontribusi terhadap model. Berikut ini prosesnya:

- Filter Methods (Statistik, Korelasi, dll.): Metode ini menggunakan teknik statistik seperti *chi-square test*, korelasi Pearson, atau ANOVA untuk menentukan fitur yang paling berkaitan dengan target.
- Wrapper Methods (Recursive Feature Elimination, Forward/Backward Selection): Menggunakan model untuk secara iteratif menilai fitur mana yang paling berpengaruh, misalnya dengan *Recursive Feature Elimination (RFE)*.
- Embedded Methods (LASSO, Decision Trees, Random Forest): Algoritma seperti *LASSO regression* atau *random forest* memiliki fitur bawaan untuk memilih fitur yang paling penting selama pelatihan model.

**e. Dimensionality Reduction (Pengurangan Dimensi)**

Dimensionality reduction atau pengurangan dimensi adalah proses mengurangi jumlah fitur atau variabel dalam dataset tanpa menghilangkan informasi penting yang terkandung di dalamnya. Dalam konteks machine learning, semakin banyak fitur yang digunakan, semakin kompleks pula model yang dibangun. Kompleksitas ini dapat menyebabkan masalah seperti meningkatnya waktu komputasi, risiko overfitting, serta kesulitan dalam interpretasi model. Oleh karena itu, pengurangan dimensi dilakukan untuk menyederhanakan data agar model dapat bekerja lebih efisien dan tetap akurat. Tujuannya bukan sekadar menghapus fitur, tetapi juga menemukan representasi baru dari data yang mampu menggambarkan pola utama secara lebih ringkas.

Jika dataset memiliki terlalu banyak fitur, teknik reduksi dimensi dapat membantu menyederhanakan data tanpa kehilangan informasi penting. Berikut prosesnya: **PCA (Principal Component Analysis):** Teknik yang mengubah fitur yang ada menjadi kombinasi baru yang lebih ringkas,

tetapi tetap mempertahankan informasi utama. **LDA (Linear Discriminant Analysis):** Berfokus pada memaksimalkan separasi antar kelas dalam dataset untuk meningkatkan klasifikasi. **t-SNE dan UMAP:** Teknik *non-linear* yang sering digunakan untuk visualisasi data berdimensi tinggi dalam bentuk dua atau tiga dimensi.

- **Tujuan Dimensionality Reduction**
  Tujuan utama pengurangan dimensi adalah untuk meningkatkan efisiensi model dan memperbaiki performa algoritma pembelajaran mesin. Dengan jumlah fitur yang lebih sedikit, model menjadi lebih mudah dilatih, lebih cepat dalam proses komputasi, dan lebih kecil kemungkinan mengalami overfitting. Selain itu, pengurangan dimensi juga membantu dalam visualisasi data, terutama ketika data memiliki dimensi yang sangat tinggi. Dengan mengubah data ke dalam dua atau tiga dimensi, pola atau kelompok dalam data dapat lebih mudah dilihat. Proses ini juga membantu dalam menghilangkan fitur-fitur yang tidak relevan atau redundan, sehingga model dapat fokus pada informasi yang benar-benar penting.

- **Masalah Dimensi Tinggi (Curse of Dimensionality)**
  Salah satu alasan utama dilakukannya pengurangan dimensi adalah adanya fenomena yang disebut *curse of dimensionality*. Ketika jumlah fitur dalam data meningkat secara signifikan, ruang data menjadi sangat besar dan jarak antar titik data juga meningkat. Hal ini membuat algoritma machine learning sulit menemukan pola karena data menjadi terlalu jarang (sparse). Sebagai akibatnya, model bisa kehilangan kemampuan generalisasi dan performanya menurun. Fenomena ini umum terjadi dalam dataset dengan ratusan atau ribuan fitur, seperti dalam data genomik, pengenalan wajah, dan pemrosesan teks. Dengan melakukan reduksi dimensi, masalah ini dapat diminimalkan dan model dapat bekerja lebih efektif.

- **Jenis-jenis Pendekatan dalam Dimensionality Reduction**
  Secara umum, terdapat dua pendekatan utama dalam pengurangan dimensi, yaitu *feature selection* dan *feature extraction*. Feature selection berfokus pada pemilihan subset dari fitur yang paling relevan terhadap target, sedangkan feature extraction menciptakan fitur baru melalui transformasi matematis dari fitur asli. Feature selection biasanya dilakukan menggunakan metode seperti *filter method* (misalnya korelasi atau uji chi-square), *wrapper method* (seperti Recursive Feature Elimination), dan *embedded method* (seperti feature importance pada Random Forest atau Lasso Regression). Sementara itu, feature extraction melibatkan teknik yang mengubah data asli ke dalam bentuk ruang baru yang berdimensi lebih rendah, contohnya Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), dan t-Distributed Stochastic Neighbor Embedding (t-SNE). Pendekatan ini tidak hanya mengurangi jumlah fitur, tetapi juga dapat mengungkapkan struktur tersembunyi dalam data.
- **Principal Component Analysis (PCA)**
  PCA adalah salah satu teknik pengurangan dimensi paling populer. Metode ini bekerja dengan mengubah kumpulan fitur asli yang saling berkorelasi menjadi serangkaian fitur baru yang tidak saling berkorelasi, disebut *principal components*. Setiap komponen utama merupakan kombinasi linear dari fitur-fitur asli dan dipilih sedemikian rupa agar dapat menangkap variasi terbesar dalam data. Komponen pertama (PC1) menjelaskan variansi terbesar, diikuti oleh komponen kedua (PC2), dan seterusnya. Dengan memilih beberapa komponen utama pertama, kita dapat mengurangi dimensi data tanpa kehilangan terlalu banyak informasi. PCA sering digunakan dalam analisis citra,

pengenalan wajah, serta pra-pemrosesan data sebelum diterapkan pada algoritma machine learning lainnya.

- **Linear Discriminant Analysis (LDA)**
  Berbeda dengan PCA yang bersifat unsupervised (tidak mempertimbangkan label kelas), LDA merupakan metode *supervised* yang mempertimbangkan informasi kelas dalam proses reduksi dimensi. Tujuan LDA adalah mencari kombinasi fitur yang memaksimalkan separasi antar kelas sambil meminimalkan variansi dalam kelas yang sama. Dengan kata lain, LDA tidak hanya mengurangi dimensi tetapi juga menjaga kemampuan diskriminatif data terhadap target. LDA sangat bermanfaat dalam masalah klasifikasi seperti pengenalan wajah, identifikasi tulisan tangan, dan analisis bioinformatika, di mana perbedaan antar kelas perlu dipertahankan secara jelas.
- **t-Distributed Stochastic Neighbor Embedding (t-SNE)**
  t-SNE merupakan teknik reduksi dimensi non-linear yang sangat efektif untuk visualisasi data berdimensi tinggi. Metode ini bekerja dengan memetakan data dari ruang dimensi tinggi ke ruang dua atau tiga dimensi sambil menjaga hubungan jarak antar data. Dengan t-SNE, data yang mirip di ruang tinggi akan tetap berdekatan setelah direduksi, sehingga sangat baik untuk melihat pola atau klaster tersembunyi dalam data. Meskipun t-SNE tidak ideal untuk pelatihan model karena hasilnya sulit direproduksi dan komputasinya berat, teknik ini sering digunakan untuk eksplorasi data atau analisis visual.
- **Autoencoder dalam Reduksi Dimensi**
  Autoencoder adalah model neural network yang digunakan untuk melakukan reduksi dimensi secara non-linear. Struktur dasarnya terdiri dari dua bagian, yaitu *encoder* yang mengubah data berdimensi tinggi menjadi representasi berdimensi rendah (latent space), dan *decoder* yang mencoba merekonstruksi data asli dari

representasi tersebut. Melalui proses pelatihan, autoencoder belajar bagaimana menyimpan informasi penting dalam bentuk yang lebih ringkas. Teknik ini banyak digunakan dalam deep learning, khususnya untuk data gambar, suara, dan teks, di mana hubungan antar fitur sering bersifat kompleks dan non-linear.

- **Manfaat dan Dampak Dimensionality Reduction**
  Proses pengurangan dimensi memiliki banyak manfaat dalam pengembangan model machine learning. Selain mempercepat pelatihan dan mengurangi kebutuhan memori, pengurangan dimensi membantu meningkatkan kinerja model dengan menghilangkan fitur yang redundan atau tidak relevan. Model yang dihasilkan menjadi lebih sederhana, lebih mudah diinterpretasikan, dan memiliki kemampuan generalisasi yang lebih baik terhadap data baru. Selain itu, dalam konteks visualisasi, reduksi dimensi membantu peneliti memahami struktur dan distribusi data secara intuitif, terutama untuk mendeteksi klaster, anomali, atau hubungan antar variabel.

## 2.3. Teknik Feature Engineering Berdasarkan Tipe Data

Setiap jenis data memiliki karakteristik unik yang memerlukan teknik *feature engineering* yang berbeda. Teknik yang tepat dapat meningkatkan pemahaman model terhadap pola dalam data dan meningkatkan akurasi prediksi. Berikut beberapa teknik *feature engineering* berdasarkan tipe data:

### 1. Numerical Data

Data numerik sering kali perlu dinormalisasi atau diubah agar lebih mudah dipahami oleh model.

- **Scaling (Min-Max, Z-score, Log Transform):** *Scaling* bertujuan untuk menyamakan skala antar fitur. Min-Max Scaling merentangkan nilai ke dalam rentang tertentu (misalnya 0-1), sedangkan Z-score *standardization* memastikan distribusi memiliki rata-rata nol dan standar deviasi satu. *Log transform* berguna untuk menangani distribusi yang miring (*skewed data*).
- **Binning (Discretization):** Teknik ini mengelompokkan nilai numerik menjadi beberapa kategori atau *bins* (misalnya, usia dikelompokkan menjadi anak-anak, remaja, dan dewasa). Ini membantu mengurangi *noise* dan membuat pola lebih jelas.

2. **Categorical Data**

Data kategorikal harus dikonversi ke bentuk numerik agar bisa digunakan dalam model *machine learning.*

- **One-hot Encoding:** Teknik ini mengubah kategori menjadi vektor biner (misalnya, kategori "merah", "biru", "hijau" menjadi tiga kolom berbeda dengan nilai 0 atau 1).
- **Label Encoding:** Setiap kategori dikonversi menjadi angka unik (misalnya, "merah" = 0, "biru" = 1, "hijau" = 2). Ini cocok untuk data dengan hubungan ordinal.
- **Target Encoding:** Mengganti kategori dengan rata-rata target berdasarkan kategori tersebut, sering digunakan dalam model berbasis statistik seperti regresi.

**3. Text Data**

Data teks membutuhkan representasi numerik agar bisa dipahami oleh algoritma *machine learning.*

- **TF-IDF (Term Frequency – Inverse Document Frequency):** Teknik ini mengukur seberapa penting sebuah kata dalam dokumen dibandingkan dengan keseluruhan kumpulan dokumen, membantu menghilangkan kata-kata yang sering muncul tetapi tidak bermakna.
- **Word Embeddings (Word2Vec, GloVe, BERT):** Representasi kata dalam bentuk vektor yang lebih kompleks, seperti *Word2Vec* dan *GloVe*, memungkinkan model memahami hubungan semantik antar kata. *BERT* lebih canggih karena mempertimbangkan konteks dalam kalimat.

**4. Time-Series Data**

Data berbasis waktu sering kali memerlukan teknik khusus untuk menangkap tren dan pola musiman.

- **Rolling Statistics (Moving Average, Exponential Smoothing):** Menggunakan rata-rata bergerak untuk menangkap tren dalam data (misalnya, rata-rata harga saham selama 7 hari terakhir).
- **Lag Features dan Seasonal Decomposition:** *Lag features* menambahkan nilai sebelumnya sebagai fitur, sementara seasonal decomposition memisahkan tren, musiman, dan komponen residu dari data waktu.

**5. Image Data**

Fitur dari gambar dapat diekstrak untuk membantu model mengenali pola visual.

- **Edge Detection:** Teknik ini menyoroti batas objek dalam gambar menggunakan algoritma seperti Canny Edge Detection.

- **Histogram of Oriented Gradients (HOG):** Metode ini mengekstrak fitur tekstur dengan menganalisis gradien dan arah tepi dalam gambar, sering digunakan dalam pengenalan objek.

6. **Geospatial Data**

Data berbasis lokasi memerlukan teknik khusus untuk memahami pola geografis.

- **Clustering Berbasis Lokasi:** Mengelompokkan titik-titik geografis berdasarkan kedekatan atau pola tertentu menggunakan algoritma seperti K-Means atau DBSCAN.
- **Distance-Based Features:** Menghitung jarak antara titik geografis tertentu, misalnya jarak rumah ke pusat kota, yang bisa menjadi fitur penting dalam model predikti.

## 2.4. Tools dan Library untuk Feature Engineering

Untuk membantu proses *feature engineering* terdapat berbagai *tools* dan *library* yang dapat digunakan, mulai dari manipulasi data hingga ekstraksi fitur menggunakan *deep learning*. Berikut beberapa *tools* dan *library* yang umum digunakan dalam *feature engineering:*

1. Pandas & NumPy untuk Manipulasi Data

Pandas dan NumPy adalah dua *library* utama dalam ekosistem Python yang digunakan untuk manipulasi data. Pandas menyediakan struktur data seperti DataFrame yang memungkinkan transformasi data dengan mudah, seperti menangani data yang hilang, melakukan *encoding* kategori, serta menggabungkan dataset. NumPy, di sisi lain, lebih berfokus pada operasi numerik

dengan *array* multidimensi yang efisien, yang sering digunakan dalam perhitungan statistik dan transformasi fitur.

2. Scikit-learn untuk Preprocessing

*Scikit-learn* menyediakan berbagai modul untuk *preprocessing data,* termasuk normalisasi, standardisasi, *encoding* kategori, dan penanganan nilai yang hilang. *Library* ini juga memiliki fitur PolynomialFeatures untuk membuat fitur baru berdasarkan kombinasi fitur yang ada serta Feature Selection untuk memilih fitur yang paling relevan dalam model *machine learning.*

3. Featuretools untuk Automated Feature Engineering

*Featuretools* adalah *library open-source* yang digunakan untuk otomatisasi *feature engineering*. Dengan konsep Deep Feature Synthesis (DFS), Featuretools mampu membuat fitur baru secara otomatis dari dataset relational, sehingga menghemat waktu dalam eksplorasi fitur. *Library* ini sangat berguna dalam menangani *data time-series* dan data berbasis entitas yang kompleks.

4. TensorFlow dan PyTorch untuk Deep Learning Feature Extraction

TensorFlow dan PyTorch merupakan dua *framework deep learning* yang dapat digunakan untuk mengekstrak fitur dari data tidak terstruktur seperti gambar, teks, dan audio. Dengan menggunakan model *deep learning* yang telah dilatih sebelumnya *(pretrained models)* seperti ResNet, BERT, atau VGG, kita dapat mengekstrak representasi fitur dari data yang kompleks dan meningkatkan performa model *machine learning* secara signifikan.

## 2.5. Studi Kasus dan Implementasi Feature Engineering

*Feature engineering* dapat diterapkan dalam berbagai jenis masalah *machine learning,* termasuk regresi, klasifikasi, NLP, dan computer vision. Berikut beberapa studi kasus dan implementasi *feature engineering* dalam berbagai skenario:

### 1. Contoh Kasus Feature Engineering dalam Regresi

Dalam masalah regresi, seperti memprediksi harga rumah, fitur-fitur yang relevan dapat meningkatkan akurasi model. Teknik *feature engineering* yang umum digunakan meliputi:

- **Transformasi variabel**: Menggunakan log transform pada harga rumah untuk mengatasi distribusi yang tidak normal.
- **Feature interaction**: Membuat fitur baru seperti *harga per meter persegi* berdasarkan luas bangunan dan harga rumah.
- **Encoding kategori**: Mengubah fitur kategori seperti tipe properti menjadi representasi numerik dengan *one-hot encoding* atau *target encoding.*

### 2. Feature Engineering dalam Klasifikasi

Dalam masalah klasifikasi, seperti mendeteksi *churn* pelanggan, teknik berikut sering diterapkan:

- **Binning**: Mengelompokkan umur pelanggan ke dalam kategori (misalnya: muda, dewasa, senior) untuk menyederhanakan hubungan dengan target.
- **Feature scaling**: Normalisasi atau standardisasi fitur numerik agar model berbasis *gradient descent* bekerja lebih optimal.

- **Feature selection**: Menggunakan mutual information atau SHAP values untuk memilih fitur yang paling berkontribusi terhadap prediksi.

**3. Feature Engineering dalam NLP dan Computer Vision**

Dalam NLP dan computer vision, *feature engineering* membantu mengubah data tidak terstruktur menjadi representasi numerik yang dapat dipahami oleh model:

- **NLP**: Menggunakan teknik seperti TF-IDF, word embeddings (Word2Vec, BERT), atau n-grams untuk menangkap hubungan antar kata dalam teks.
- **Computer Vision**: Ekstraksi fitur dari gambar menggunakan HOG (Histogram of Oriented Gradients), SIFT (Scale-Invariant Feature Transform), atau menggunakan *pretrained deep learning* models untuk mendapatkan fitur tingkat tinggi dari gambaar.

# BAB III

# PENUTUP

## 3.1. Kesimpulan

Pra-pemrosesan data merupakan tahap fundamental dalam proses *machine learning* karena menentukan kualitas data yang akan digunakan untuk pelatihan model. Melalui *feature engineering*, data mentah dapat diubah menjadi fitur-fitur yang lebih informatif dan representatif, sehingga membantu model memahami pola dengan lebih baik. Proses *feature selection* berperan dalam menyaring fitur yang paling relevan dan menghapus fitur yang tidak penting, guna meningkatkan efisiensi serta mengurangi risiko *overfitting*. Sementara itu, *dimensionality reduction* atau pengurangan dimensi bertujuan untuk menyederhanakan data tanpa menghilangkan informasi penting. Teknik seperti *Principal Component Analysis (PCA)* mampu mereduksi fitur ke dalam bentuk yang lebih ringkas dengan tetap mempertahankan variasi terbesar dari data asli. Penerapan *feature selection* dan *dimensionality reduction* secara bersamaan memungkinkan model *machine learning* bekerja lebih cepat, akurat, dan efisien. Secara keseluruhan, kombinasi antara *feature engineering*, *feature selection*, dan *dimensionality reduction* menjadi langkah strategis dalam membangun model pembelajaran mesin yang optimal, terukur, dan mudah diinterpretasikan.

## 3.2. Saran

Dalam penerapan *feature engineering* dan *dimensionality reduction*, diperlukan pemahaman yang mendalam terhadap karakteristik data serta konteks permasalahan yang dihadapi. Pemilihan metode yang tepat, seperti penggunaan PCA atau teknik seleksi fitur berbasis model, harus disesuaikan dengan jenis data dan tujuan analisis. Peneliti dan praktisi disarankan untuk

terus mengembangkan keterampilan dalam eksplorasi fitur serta memanfaatkan *tools* modern seperti Scikit-learn, Featuretools, dan TensorFlow untuk mempercepat proses rekayasa fitur.