**Rangkuman Dasar-Dasar Analisis Data**

**Pengenalan Analisis Data**

Data merupakan sekumpulan fakta yang dapat direpresentasikan dalam berbagai bentuk seperti angka, gambar, video, teks, hasil pengukuran, dll. Sebagai generasi kekinian, kita telah memproduksi banyak data setiap harinya. Selain menghasilkan banyak data, tanpa sadar kita juga sering memanfaatkan data untuk mengambil sebuah keputusan.

Analisis data dapat diartikan sebagai sebuah proses mengumpulkan, mentransformasi, dan menata data untuk menarik kesimpulan, membuat prediksi, serta memberi pertimbangan yang tepat dalam mengambil keputusan. Orang yang melakukan proses analisis data ini sering disebut sebagai *data analyst*.

Untuk mendukung proses analisis data, seorang data analyst membutuhkan beberapa keterampilan seperti berikut.

* **Analytical Skills**  
  Keterampilan ini berhubungan dengan kualitas dan karakteristik yang berhubungan dengan penyelesaian masalah menggunakan fakta. Berikut merupakan beberapa poin penting dalam analytical skill.
  + Memiliki rasa ingin tahu yang tinggi untuk belajar hal baru serta mencari pengalaman dan tantangan yang baru.
  + Memiliki kemampuan untuk memahami konteks atau kondisi di balik suatu kejadian.
  + Memiliki pola pikir teknis sehingga mampu menguraikan suatu hal menjadi langkah-langkah yg lebih kecil dan mengerjakan tiap bagiannya dengan cara yang teratur dan logis.
  + Mampu mengolah atau mengatur informasi secara baik dengan menerapkan *data design*.
  + Mampu menerapkan *data strategy*untuk manajemen orang, proses, dan tools yang digunakan dalam analisis data.
* **Analytical Thinking**  
  Hal ini melibatkan pengidentifikasian dan pendefinisian sebuah masalah, kemudian menyelesaikannya berdasarkan data dengan cara yang terorganisasi dan berurutan. Secara umum, terdapat enam aspek kunci dari analytical thinking antara lain sebagai berikut.
  + Merepresentasikan sebuah informasi dalam bentuk grafik atau visual. Hal ini akan membantu seorang data analyst untuk memahami dan menyampaikan sebuah informasi secara efektif.
  + Selalu menerapkan pola pikir strategis untuk menjaga fokus dengan membuat perencanaan matang.
  + Menerapkan pola pikir yang berorientasi pada masalah untuk mengidentifikasi, mendeskripsikan, dan menyelesaikan sebuah *business problem*.
  + Mengidentifikasi korelasi antara dua data atau lebih.
  + Menerapkan *big-picture thinking* (pemikiran gambaran besar) dalam proses analisis.
  + Mampu menerapkan *detail-oriented thinking* (pemikiran yang berorientasi pada detail) dalam proses analisis.

**Pentingnya Data dalam Dunia Bisnis**

Di zaman yang serba canggih seperti saat ini, terdapat banyak inovasi dalam pemanfaatan teknologi dan informasi terutama pada bidang data. Data merupakan salah satu resource penting bagi perusahaan dari berbagai industri mulai dari *e-commerce*, *entertainment*, *manufacturing*,*healthcare*, *marketing*, *finance*, *tech*, dll.

Berbagai perusahaan tersebut membutuhkan data untuk mengidentifikasi peluang, tren, mengoptimalkan proses yang ada, meningkatkan layanan pelanggan, dan masih banyak lagi penerapannya di industri. Hal inilah yang mengakibatkan data dijuluki sebagai “*The new oil*” karena dianggap sebagai sumber daya paling penting saat ini.

Selain itu, *insigh*t (wawasan) yang diperoleh dari data dapat digunakan sebagai pendukung dalam pengambilan keputusan. Praktek seperti ini mendorong terbentuknya sebuah kultur baru dalam dunia bisnis yang disebut *data-driven decision making*.

Data-driven decision making merupakan sebuah kultur, *best practice*, serta proses dalam penggunaan data dan hasil analisis untuk memandu seluruh pengambilan keputusan bisnis yang tentunya sejalan dengan objektif perusahaan.

Umumnya perusahaan atau organisasi yang menerapkan kultur ini memiliki beberapa karakteristik seperti berikut.

* Memiliki data yang tersentralisasi dan terorganisir dengan baik.
* Memiliki kebijakan dan ketentuan terkait tata kelola data.
* Memiliki tingkat aksesibilitas data yang tinggi.
* Memiliki tool analisis yang mudah digunakan.

Berikut merupakan beberapa manfaat penerapan data-driven decision making.

* Manfaat utama dalam data-driven decision making ialah akuntabilitas dan konsistensi dalam setiap pengambilan keputusan. Hal ini akan mempermudah perusahaan dalam membuat sebuah *benchmark* yang jelas untuk mengevaluasi setiap keputusan dan strategi bisnis yang dibuat sehingga mampu mendorong perkembangan perusahaan secara kontinu.
* Pengambilan keputusan berdasarkan data akan mengarahkan perusahaan untuk berinovasi dan memperoleh peluang bisnis.
* Data-driven decision making mampu meningkatkan efisiensi dengan menyediakan berbagai informasi yang relevan dalam mengambil sebuah keputusan.

Terdapat beberapa poin yang harus diperhatikan ketika ingin menerapkan kultur data-driven decision making yaitu seperti berikut.

* Mengumpulkan data sebanyak mungkin
* Menentukan tujuan yang dapat diukur
* Memastikan Aksesibilitas Data
* Menentukan prioritas

**Mengenal Siklus Analisis Data**

Seperti yang telah kita bahas bersama, analisis data merupakan sebuah proses mengumpulkan, mentransformasi, dan menata data untuk menarik kesimpulan, membuat prediksi, serta memberi pertimbangan yang tepat dalam mengambil keputusan. Sebagai suatu proses, analisis data tentunya memiliki beberapa tahapan seperti berikut.

1. **Mendefinisikan pertanyaan**  
   Pada awal proyek analisis data, kita perlu mendefinisikan berbagai pertanyaan analisis terkait data ataupun permasalahan yang ingin diselesaikan. Pertanyaan tersebut akan membantu kita dalam memahami masalah bisnis yang dihadapi serta tujuan yang ingin dicapai. Bukan hanya itu, dengan adanya pertanyaan tersebut, kita akan merasa terbantu karena bisa lebih fokus pada bagian data yang relevan dengan tujuan.  
     
   Pada proses analisis data, terkadang kita diberikan data terlebih dahulu lalu mulai mendefinisikan pertanyaan analisis berdasarkan data tersebut. Namun, tidak jarang kita juga berangkat dari pertanyaan analisis terlebih dahulu lalu mencari data yang sesuai untuk menjawab pertanyaan tersebut.
2. **Data wrangling**  
   Setelah kita memiliki pertanyaan analisis yang baik, tahap selanjutnya adalah data wrangling. Pada tahap ini, Anda memulai dengan mengumpulkan data yang dibutuhkan untuk menjawab pertanyaan analisis yang telah dibuat sebelumnya. Kemudian Anda perlu menilai kualitas dan struktur dari data tersebut. Hal ini dilakukan untuk mengidentifikasi masalah yang terdapat dalam data serta membuat strategi pembersihan data yang sesuai.  
     
   Proses pembersihan data umumnya dilakukan dengan memodifikasi, mengganti, atau menghilangkan data yang bermasalah sehingga diperoleh dataset yang berkualitas dan terstruktur dengan baik.*FYI*, pembersihan data merupakan salah satu proses yang paling menghabiskan waktu dalam proyek analisis data. Oleh karena itu, kita sangat disarankan untuk menyediakan waktu yang cukup untuk melakukan pembersihan data.
3. **Exploratory Data Analysis**  
   Tahap berikutnya adalah data exploratory data analysis. Pada tahap inilah Anda akan melakukan eksplorasi terhadap data yang telah dibersihkan untuk memperoleh *insight* dan menjawab pertanyaan analisis.  
     
   Proses eksplorasi umumnya dilakukan dengan memanfaatkan berbagai teknik *descriptive statistics* (Penggunaan konsep statistik untuk mendeskripsikan data). Hal ini bertujuan untuk menemukan pola, hubungan, serta membangun intuisi terkait data yang diolah.  
     
   Selain menggunakan descriptive statistic, terkadang kita juga menggunakan teknik machine learning atau *inferential statistic* untuk memprediksi data di masa depan. Namun, kedua teknik tersebut di luar cakupan pembahasan kita pada kelas ini.
4. **Data visualization**  
   Hasil yang diperoleh dari exploratory data analysis perlu divisualisasikan melalui tahap data visualization. Pada tahap ini, kita akan menerapkan berbagai teknik dan jenis visualisasi data yang tepat untuk mengomunikasikan temuan atau insight dari hasil analisis secara efektif.
5. **Draw conclusion & communicate**  
   Pada akhir proyek analisis data, kita perlu membuat conclusion atau kesimpulan dari hasil analisis. Kesimpulan yang dibuat haruslah menjawab semua pertanyaan dan tujuan yang telah didefinisikan di awal.  
     
   Selain membuat kesimpulan, kita juga perlu membagikan temuan atau insight dari hasil analisis. Terdapat banyak cara dalam mengomunikasikan hasil analisis, seperti menggunakan report, slide deck, dan atau presentation.

Di industri, tahapan tersebut tidak selamanya berbentuk liner melainkan ada beberapa tahapan yang berulang. Hal inilah yang membuat tahapan dalam analisis data lebih cocok disebut sebagai sebuah siklus karena terdapat banyak iteratif proses di dalamnya.

**Pengenalan Tool Pengolahan Data**

Sebagai salah satu bahasa pemrograman kekinian, Python dapat digunakan untuk mengerjakan banyak tugas salah satunya adalah mengolah dan memvisualisasikan data.

Untuk mendukung proses pengolahan dan analisis data, Python memiliki banyak *libraries* dan *tools* yang bersifat *open-source* dan dapat digunakan secara bebas. Berikut merupakan tiga libraries yang paling sering digunakan untuk mengolah data.

* **Library NumPy**  
  NumPy (singkatan dari *numerical Python*) merupakan salah satu library Python yang sangat *powerful* untuk membuat dan mengolah *multi-dimensional arrays* (sering juga disebut sebagai matriks atau tensor). NumPy dibuat dengan tujuan untuk mempercepat proses komputasi numerik dalam bahasa pemrograman Python.
* **Library pandas**  
  Selain NumPy, library lain yang sering digunakan adalah pandas. Ia merupakan library Python yang spesifik digunakan untuk memanipulasi dan menganalisis data. Library inilah yang akan kita gunakan sebagai tool utama dalam menganalisis data.
* **Library SciPy**  
  Library penting selanjutnya adalah SciPy (singkatan dari Scientific Python). Ia merupakan library Python yang khusus digunakan untuk kebutuhan komputasi saintifik. Library ini memanfaatkan NumPy sebagai komponen dasar dalam setiap perhitungan komputasi yang akan dilakukan.

**Pengenalan Tool Visualisasi Data**

Pada proses analisis data, selain mengolah data, kita juga perlu merepresentasikan data tersebut ke dalam bentuk visual yang menarik dan mudah dipahami. Nah, pada materi kali ini, kita akan berkenalan dengan berbagai tools visualisasi data yang ada dalam bahasa pemrograman Python.

Sebagai salah satu bahasa pemrograman paling populer di kalangan praktisi data, Python memiliki banyak *libraries* dan *tools* yang bersifat*open-source* untuk mendukung pembuatan visualisasi data yang baik dan menarik. Berikut merupakan dua library Python yang paling sering digunakan oleh praktisi data di industri.

* **Library matplotlib**  
  Matplotlib merupakan sebuah library Python yang *reliable* dan komprehensif untuk mendukung kebutuhan pembuatan visualisasi data. Matplotlib menyediakan banyak *object* dan *method* yang memungkinkan kita membuat visualisasi data statis, *animated*, dan interaktif.
* **Library seaborn**  
  Selain matplotlib, library lain yang sering digunakan untuk visualisasi data adalah seaborn. Seaborn merupakan library Python yang spesifik digunakan untuk membuat visualisasi data yang atraktif dan informatif. Library ini memanfaatkan matplotlib untuk menampilkan grafik visualisasi data.

## Rangkuman Penerapan Dasar-Dasar Descriptive Statistics

### Pengantar Penerapan Dasar-Dasar Statistik

By definition, statistik merupakan ilmu yang menerapkan berbagai metode saintifik dalam proses pengumpulan, analisis, interpretasi, serta penyajian data. Terdapat beberapa konsep statistik yang umum digunakan untuk mendeskripsikan sebuah data yang dikenal sebagai descriptive statistics. Beberapa konsep tersebut bisa Anda lihat di bawah ini.

* Measuring central tendency untuk mengidentifikasi nilai khas atau sentral dalam sebuah data.
* Measuring dispersion untuk menilai variabilitas atau sebaran nilai dalam sebuah data.
* Measuring asymmetric untuk menilai ketidaksimetrisan dalam sebuah distribusi data.
* Data relationship untuk mengidentifikasi hubungan dua atau lebih feature/column/variable dalam sebuah data.

#### Tipe Data

Pada proses analisis, tipe data akan menentukan metode analisis dan bentuk visualisasi data yang tepat. Secara garis besar, data dapat dibagi menjadi dua tipe, yaitu kuantitatif (numerik) dan kategoris.

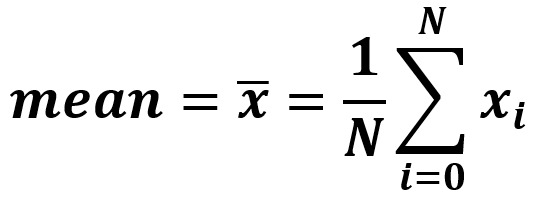
|  |  |  |  |
| --- | --- | --- | --- |
| Data kuantitatif | Data kontinu | Direpresentasikan dalam berbagai nilai numerik dan bisa dibagi atau diubah ke dalam unit yang lebih kecil | Umur, tinggi badan, berat badan, dll. |
| Data diskret | Direpresentasikan dalam bilangan bulat dan tidak dapat dibagi ke dalam unit yang lebih kecil | Jumlah kucing peliharaan, jumlah produk yang terjual, dll. |
| Data kategoris | Data ordinal | Tipe data kategoris yang dapat diurutkan berdasarkan rangking | Rating, kategori pelanggan berdasarkan income, dll. |
| Data nominal | Tipe data kategoris yang tidak memiliki urutan atau rangking | Jenis kelamin, warna bulu kucing, dll. |

### Measuring Central Tendency

Measuring central tendency merupakan sekumpulan parameter statistik untuk menggambarkan nilai khas atau sentral yang mewakili keseluruhan observasi atau data. Nilai khas atau sentral tersebut direpresentasikan menggunakan tiga parameter statistik yaitu mean, median, dan mode.

#### Mean

Mean atau sering juga disebut sebagai expected value merupakan salah satu parameter yang paling sering digunakan untuk menggambarkan nilai khas atau sentral dari suatu data. Ia diperoleh dari menghitung jumlah keseluruhan data dan dibagi dengan banyaknya data yang dimiliki.

[](https://www.dicoding.com/academies/555/tutorials/30370?from=30365)

Parameter ini sangat sensitif terhadap noise dan outlier yang terdapat dalam sebuah data.  Inilah salah satu alasan kenapa kita tidak boleh hanya menggunakan mean untuk merepresentasikan nilai khas dari suatu data.

#### Median

Median merupakan parameter yang merepresentasikan nilai tengah atau persentil ke-50 dari keseluruhan observasi atau data. Parameter ini diperoleh dengan mengambil nilai persentil ke-50 dari data yang telah diurutkan.

Untuk data yang berjumlah ganjil, nilai median diperoleh dengan mengambil langsung nilai tengah dari data yang telah diurutkan. Namun, hal ini berbeda jika data yang dimiliki berjumlah genap. Pada kasus seperti ini, kita perlu mengurutkan dan menghitung nilai rata-rata dari dua data yang berada di tengah

Median dikenal sebagai salah satu parameter yang sangat toleran terhadap outlier. Apabila Anda mengubah salah satu nilai dari data tersebut, parameter median yang diperoleh tidak akan berubah secara signifikan.

#### Mode

Parameter ketiga dalam measuring central tendency ialah mode. Parameter ini akan memberikan kita gambaran atau informasi terkait nilai yang paling sering muncul dalam suatu data.

Untuk menghitung parameter mode, kita perlu mengurutkan nilai dari suatu data terlebih dahulu. Selanjutnya memilih nilai yang paling sering muncul. Sebagai contoh, pada data jumlah kucing peliharaan yang dimiliki pegawai Dicoding, nilai yang paling sering muncul adalah 2. Nilai ini merupakan mode dari data tersebut.

Ketika menggunakan mode, Anda perlu memahami bahwa parameter ini akan mencari nilai yang paling sering muncul. Oleh karena itu, jika suatu data memiliki nilai yang unik, data tersebut tidak memiliki mode. Selain itu, apabila data memiliki lebih dari satu nilai yang paling sering muncul, itu artinya data tersebut memiliki lebih dari satu mode.

### Measuring Dispersion

Pada materi kali ini, kita akan fokus membahas sebuah konsep statistik untuk menilai sebaran dari suatu data yaitu measuring dispersion atau sering juga disebut measures of spread. Ia akan memberikan informasi terkait seberapa jauh suatu titik data dengan titik data lainnya. Hal ini mampu memberikan kita gambaran variasi nilai dari data tersebut. Untuk menggambarkan measuring dispersion, kita dapat menggunakan empat parameter, yaitu range, interquartile range, variance, dan standard deviation.

#### Range

Parameter pertama yang akan kita bahas adalah range. Ia merupakan parameter yang digunakan untuk melihat perbedaan antara nilai maksimum dan minimum dari suatu data.

Untuk menghitung parameter ini, kita perlu mengidentifikasi nilai maksimum dan minimum dari suatu data terlebih dahulu. Selanjutnya, kita mengurangi nilai maksimum dan nilai minimum. Selisih inilah yang menjadi parameter range.

#### Interquartile Range

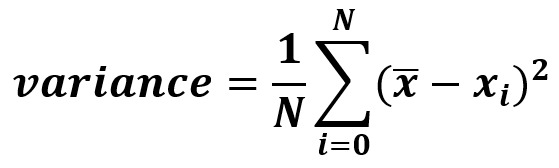
Interquartile Range atau sering disingkat IQR merupakan parameter statistik yang menggambarkan selisih antara kuartil ketiga (Q3) dan kuartil pertama (Q1). Parameter ini mewakili nilai range atau rentang di mana sebagian besar titik data berada.

Untuk menghitung IQR, kita perlu menentukan nilai Q1 dan Q3 terlebih dahulu. Selanjutnya, kita dapat menghitung selisih keduanya. Selisih inilah yang menjadi nilai dari parameter IQR.

#### Variance

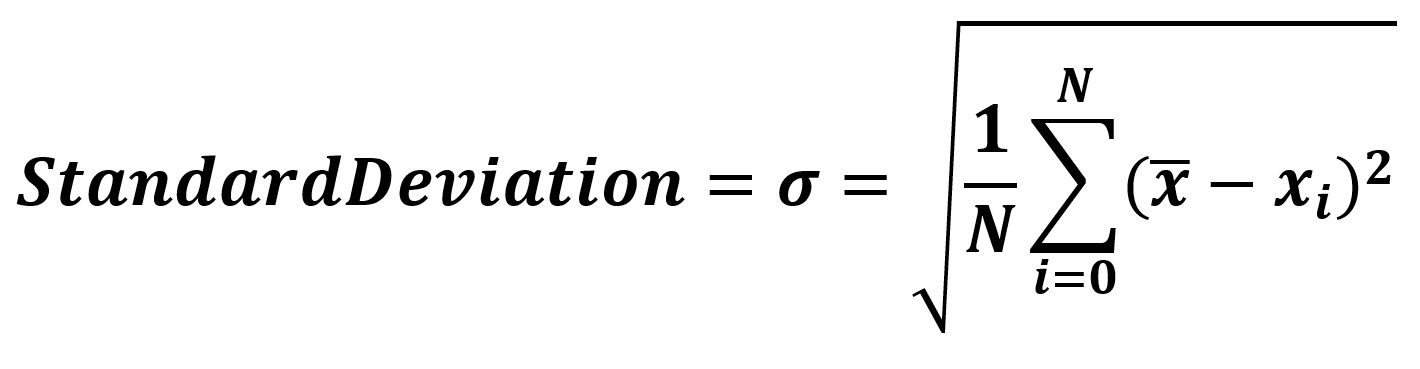
Selain range dan IQR, sebaran dari suatu data juga dapat direpresentasikan menggunakan parameter variance. Ia merupakan parameter yang digunakan untuk menggambarkan besar simpangan suatu titik data dari nilai mean-nya.

Sebelum menghitung nilai variance, kita membutuhkan nilai mean terlebih dahulu. Berikutnya kita perlu menghitung total selisih kuadrat dari seluruh titik data dengan nilai mean-nya. Nilai total tersebut selanjutnya akan dibagi dengan banyak titik data yang dimiliki. Berikut merupakan rumus untuk menghitung variance.

[](https://www.dicoding.com/academies/555/tutorials/30370?from=30365)

#### Standard Deviation

Jika Anda perhatikan, pada perhitungan variance kita sebenarnya telah menghitung rata-rata selisih kuadrat antar titik data dan nilai mean-nya. Penggunaan selisih kuadrat ini membuat unit dari parameter variance berbeda dengan unit data sehingga cukup sulit untuk diinterpretasikan. Oleh karena itu, kita membutuhkan satu parameter lain yang akan menghitung akar kuadrat dari variance. Parameter tersebut bernama standard deviation.

[](https://www.dicoding.com/academies/555/tutorials/30370?from=30365)

Standard deviation merupakan parameter yang paling sering digunakan para praktisi data untuk menilai sebaran atau simpangan dalam sebuah data. Semakin kecil nilai standard deviation, semakin kecil pula jarak antar titik data dengan nilai mean-nya sehingga bisa disimpulkan data tersebut memiliki sebaran yang sempit. Di sisi lain, semakin besar nilai standard deviation, semakin luas pula sebaran datanya. Selain itu, standard deviation juga sering digunakan untuk menilai seberapa baik parameter mean dalam merepresentasikan suatu data.

### Measuring Asymmetric

Measuring asymmetric merupakan konsep statistik yang digunakan untuk mengidentifikasi ketidaksimetrisan dalam sebuah distribusi data numerik. Parameter statistik yang umum digunakan dalam measuring asymmetric ialah skewness.

#### Distribusi Data

Distribusi data atau data distribution merupakan sebuah konsep statistik yang digunakan untuk menunjukkan frekuensi suatu nilai muncul dalam sebuah data. Umumnya kita menggunakan grafik histogram untuk melihat distribusi suatu data. Grafik tersebut akan menunjukkan frekuensi kemunculan setiap nilai dalam sebuah data.

Secara umum, berdasarkan bentuk grafik histogram yang terbentuk, kita bisa membagi distribusi data dalam tiga bentuk yaitu symmetric distribution, right-skewed distribution, dan left-skewed distribution.

* **Symmetric distribution**  
  Symmetric distribution (normal distribution atau bell curve distribution) merupakan suatu distribusi data yang memungkinkan kita untuk membagi data menjadi dua bagian secara simetris. Distribusi data ini memiliki nilai mean, median, serta mode yang sama dan berada di sentral distribusi data.
* **Right-skewed distribution**  
  Right-skewed distribution merupakan distribusi data yang memiliki sebagian besar populasi data yang terkonsentrasi pada bagian kiri. Distribusi data ini memiliki nilai mean lebih besar dari nilai median dan juga mode.
* **Left-skewed distribution**  
  Left-skewed distribution merupakan distribusi data yang terjadi ketika sebagian besar populasi data berada pada bagian kanan. Umumnya distribusi ini memiliki nilai median dan mode yang lebih besar dari nilai mean.

#### Skewness

Skewness merupakan parameter statistik yang digunakan untuk mengukur kesimetrisan sebuah distribusi data. Ia mampu memberikan kita gambaran tentang banyaknya data yang menyimpang dari symmetric distribution. Parameter ini memiliki jangkauan nilai mulai dari negatif hingga positif.

* Nilai skewness positif menggambarkan data yang memiliki distribusi yang cenderung right-skewed.
* Nilai skewness nol menggambarkan data yang memiliki distribusi simetris sempurna.
* Nilai skewness negatif merepresentasikan data dengan distribusi yang cenderung left-skewed.

### Data Relationship

By definition, data relationship merupakan konsep statistik untuk mengidentifikasi hubungan antar feature bertipe numerik dalam sebuah data. Untuk mengidentifikasi hubungan antar feature bertipe numerik, kita bisa menggunakan dua parameter yaitu correlation dan covariance.

#### Correlation

Parameter pertama yang akan kita bahas untuk mengidentifikasi data relationship ialah correlation. Parameter ini digunakan untuk mengidentifikasi korelasi atau hubungan dari dua feature numerik dalam sebuah data. Korelasi ini digambarkan menggunakan nilai dengan rentang -1 hingga 1.

Pada parameter correlation, nilai negatif menggambarkan korelasi berlawanan, sedangkan nilai positif merepresentasikan korelasi bersesuaian. Jika correlation dari dua feature bernilai nol, keduanya dinyatakan tidak memiliki korelasi.

#### Covariance

Selain menggunakan correlation, kita juga bisa menggunakan parameter covariance untuk mengidentifikasi hubungan antar dua feature dalam sebuah dataset. Berdasarkan nilainya, covariance terbagi menjadi tiga bagian seperti berikut.

* Positive covariance yang menggambarkan dua feature yang berkorelasi positif atau bersesuaian.
* Negative covariance yang merepresentasikan dua feature yang berkorelasi negatif atau berlawanan.
* Zero covariance yang menandakan dua feature yang tidak berkorelasi satu sama lain.

**Rangkuman Petimbangan dalam Pengolahan Data**

**Pengantar Petimbangan dalam Pengolahan Data**

Mungkin sebagian dari Anda pernah mendengar tentang beberapa dataset akademik yang umum digunakan sebagai bahan latihan para calon praktisi data masa depan, seperti *iris dataset*, *titanic dataset*, dll. Seluruh dataset tersebut memang dikhususkan sebagai bahan latihan dan *benchmark* dalam penelitian sehingga bentuknya sangat sederhana. Selain itu, dataset seperti ini umumnya sudah dalam keadaan bersih dan siap untuk digunakan dalam proses analisis data ataupun proses lanjutan lainnya.

Data yang kita jumpai di industri sangatlah kompleks dan umumnya masih sangat kotor sehingga perlu dilakukan tahap pembersihan data (data cleaning) terlebih dahulu sebelum digunakan. Selain itu, pada proses analisis data, kita sering kali harus menggunakan lebih dari satu dataset dengan ukuran data yang sangat besar untuk menjawab sebuah masalah bisnis.

Tantangan lain yang akan Anda jumpai di industri salah satunya adalah kompleksitas dari kebutuhan bisnis. Kebutuhan bisnis di industri umumnya dipengaruhi banyak hal, mulai dari keadaan pasar, keberadaan kompetitor, hingga keadaan makro ekonomi dan tentunya masih banyak lagi faktor yang lain. Semua faktor tersebut membuat kebutuhan bisnis di industri menjadi sangat dinamis.

Selain tantangan terkait kebutuhan bisnis dan data, seorang praktisi data juga dihadapkan dengan banyak tantangan lain seperti privasi dan etika dalam pengolahan data serta keamanan dari data itu sendiri. Saat ini, ada banyak regulasi yang mengatur kedua hal tersebut dan sebagai calon praktisi data yang Andal, kita harus mematuhi semua regulasi tersebut.

**Bias dalam Data**

*By definition*, bias merupakan sebuah prasangka dan kecenderungan yang mendukung atau menentang suatu hal, individu, atau kelompok lain dengan cara yang kurang adil. Ia bisa terjadi secara sengaja maupun tidak sengaja. Bias merupakan salah satu masalah yang sering dijumpai di kehidupan sehari-hari termasuk dalam sebuah data.

Bias dalam data umumnya didefinisikan sebagai eror atau galat yang secara sistematis mengarahkan hasil menuju arah tertentu. Bias dalam data umumnya muncul disebabkan banyak faktor salah satunya ialah kesalahan dalam proses sampling.

Secara umum bias dalam data dapat dikategorikan dalam empat tipe, yaitu *sampling bias*, *observer bias*,*interpretation bias*, dan *confirmation bias*.

* **Sampling bias**  
  Sampling bias terjadi ketika sampel tidak mewakili populasi secara keseluruhan. Ini merupakan salah satu tipe bias yang paling sering terjadi. Bias seperti ini dapat dihindari menggunakan teknik *random sampling* ketika mengumpulkan sebuah data. Teknik ini memungkinkan semua bagian dari populasi mendapat kesempatan yang sama untuk terpilih.
* **Observer bias**  
  Tipe bias selanjutnya ialah observer bias atau sering juga disebut sebagai *experimenter bias*. Bias tipe ini muncul karena adanya kecenderungan yang berbeda dari setiap individu ketika melakukan observasi. Hal ini bisa dihindari dengan melakukan observasi berulang. Sebagai contoh, untuk menghindari observasi bias ketika melakukan pengukuran, kita bisa melakukan proses pengukuran secara berulang dan mengambil nilai rata-ratanya sebagai hasil akhir.
* **Interpretation bias**  
  Interpretation bias merupakan bias yang terjadi karena adanya kecenderungan kita dalam menginterpretasikan situasi ambigu hanya ke dalam dua keadaan (hitam dan putih). Hal ini berpotensi melahirkan bias karena setiap individu memiliki kecenderungan yang berbeda dalam menginterpretasi situasi yang ambigu. Hal ini dapat dihindari dengan cara melibatkan orang lain untuk mengonfirmasi hasil interpretasi yang kita buat.
* **Confirmation bias**  
  Tipe bias terakhir adalah confirmation bias. Bias ini muncul karena adanya kecenderungan kita dalam mencari atau menafsirkan informasi untuk mengonfirmasi keyakinan yang sudah ada sebelumnya. Tipe bias ini akan sangat berbahaya karena dapat menjebak kita dalam *echo chamber*. Oleh karena itu, sebagai calon praktisi data yang andal, kita harus selalu menempatkan diri senetral mungkin dalam setiap proses analisis data.

**Data Credibility**

Kredibilitas data merupakan bagian dari *data quality assessment* yang digunakan sebagai sebuah tolok ukur untuk menilai tingkat kepercayaan terhadap suatu data.

Tingkat kepercayaan dari sebuah data akan sangat berdampak terhadap kualitas dan keakuratan hasil analisis yang akan dibuat. Tingkat kepercayaan ini tentunya dipengaruhi oleh banyak faktor, seperti sumber data, kelengkapan data, keakuratan data, dll.

Berikut merupakan beberapa poin sebagai patokan untuk memastikan data yang digunakan berkualitas dan dapat dipercaya.

* **Reliabilitas data**  
  Reliabilitas merupakan salah satu faktor penting untuk menilai kualitas dan kredibilitas dari sebuah data. Data disebut reliabel jika memuat informasi yang akurat, lengkap, dan tidak bias.
* **Orisinalitas data**  
  Selain reliabilitas, orisinalitas dari sebuah data juga merupakan komponen kunci dalam menilai kualitas dan kredibilitas dari sebuah data. Hal ini erat kaitannya dengan *ownership* (kepemilikan) dan sumber data.
* **Komprehensif**  
  Poin selanjutnya yang tidak kalah penting ialah komprehensif. Poin ini sangat penting untuk memastikan data yang digunakan memiliki semua informasi yang dibutuhkan untuk menjawab pertanyaan dan tantangan bisnis.
* **Aktualitas data**  
  *Last*,*but not least* adalah aktualitas data. Seperti yang telah kita bahas sebelumnya, data di industri umumnya bersifat dinamis atau terus berubah. Jika kita menggunakan data lama, informasi di dalamnya mungkin akan *outdated* sehingga berdampak terhadap hasil analisis yang dibuat. Oleh karena itu, sangat penting bagi kita untuk memastikan proses analisis dilakukan menggunakan data terbaru.

**Privasi dan Etika dalam Pengolahan Data**

Pada beberapa dekade terakhir, kedua topik tersebut telah menjadi salah satu pembahasan hangat dalam dunia data. Hal ini merupakan imbas dari maraknya pemanfaatan data di industri sehingga mulai banyak yang menyadari potensi bahaya dalam praktik tersebut seperti pelanggaran privasi.

Untuk mencegah bahaya tersebut, mulai muncul banyak studi terkait *data ethics*. Ia merupakan kumpulan acuan standar dalam menilai benar dan salah sebuah proses pengolahan data. Berikut merupakan beberapa aspek penting dalam data ethics.

* **Ownership**  
  Aspek ini menyatakan bahwa kepemilikan data berapa pada setiap individu yang menghasilkan data tersebut. Sebagai pemilik data, individu memiliki kontrol penuh akan seluruh proses dan penggunaan data tersebut.
* **Transparency**  
  Seluruh tahapan dan algoritma dalam proses pengolahan data harus transparan dan dipahami oleh setiap individu yang menghasilkan data tersebut.
* **Consent**  
  Setiap individu harus mengetahui secara eksplisit terkait penggunaan dan alasan suatu data dikumpulkan. Hal ini dilakukan sebelum individu tersebut menyetujui pengumpulan data tersebut.
* **Currency**  
  Jika data digunakan untuk transaksi keuangan, setiap individu yang menghasilkan data harus mengetahui dengan jelas detail terkait transaksi tersebut.
* **Privacy & Openness**  
  Organisasi yang mengumpulkan data bertanggung jawab dalam menjaga setiap informasi yang bersifat pribadi pada setiap proses pengolahan data. Selain itu, organisasi tersebut harus menyediakan akses terhadap penggunaan dan pemanfaatan data (non private) secara bebas.

Berikut merupakan beberapa regulasi yang dibuat untuk menjaga data privasi milik pengguna dan memastikan penerapan data ethics.

* General Data Protection Regulation (GDPR)
* Peraturan Menteri (Permen) No 20 Tahun 2016 tentang Perlindungan Data Pribadi (PDP)

**Keamanan Data**

Data security dapat didefinisikan sebagai upaya dalam menjaga dan melindungi informasi digital (data) dari berbagai pihak yang tidak bertanggung jawab dalam seluruh proses pengolahannya. Hal ini mencakup keamanan fisik hardware (storage device) serta keamanan dari software atau aplikasi yang bersangkutan. Selain itu, ia juga mencakup seluruh kebijakan dan prosedur dalam sebuah organisasi.

Berikut merupakan dua teknik atau metode A.

* **Encryption**  
  Encryption merupakan teknik yang memanfaatkan algoritma tertentu untuk mengubah data menjadi bentuk *unusable* bagi individu atau aplikasi yang tidak memiliki akses terhadap algoritma tersebut. Pada penerapnnya, kita bisa menganggap algoritma tersebut sebagai *key* (kunci) untuk membalikkan (*reverse*) proses encryption.
* **Tokenization**  
  Metode ini memungkinkan kita untuk mengubah elemen tertentu dalam sebuah data menjadi sebuah data random yang berperan sebagai token. Pada metode ini, data asli beserta *token mapping* akan disimpan dalam lokasi yang berbeda. Hal ini memungkinkan kita membatasi akses terhadap data asli dan memastikan keamanannya dengan menyimpan data tersebut di dalam lokasi yang berbeda.

## Rangkuman Data Wrangling

### Pengenalan Data Wrangling

Data wrangling merupakan sebuah proses atau kumpulan kegiatan yang meliputi pengumpulan data (Gathering data), penilaian data (Assessing data), serta pembersihan data (Cleaning data) sebelum data digunakan dalam proses analisis data.

* **Gathering data**  
  Tahap data wrangling, dimulai dengan proses pengumpulan data. Pada proses ini kita akan mengumpulkan semua data yang dibutuhkan untuk menjawab semua pertanyaan atau masalah bisnis yang ingin kita hadapi.
* **Assessing data**  
  Setelah semua data yang dibutuhkan terkumpul, proses selanjutnya ialah penilaian terhadap data tersebut. Proses ini dilakukan untuk menilai kualitas dan struktur dari sebuah data. Selain itu, proses ini juga bertujuan untuk mengidentifikasi berbagai masalah yang terdapat dalam data, seperti missing value, unstandard value, dll.
* **Cleaning data**  
  Apabila pada proses sebelumnya kita menemukan masalah (missing value, outlier, dll.) yang terdapat di dalam sebuah data, masalah tersebut harus dibersihkan sebelum masuk tahap analisis data. Terdapat beberapa teknik yang dapat kita gunakan untuk membersihkan data. Seluruh teknik tersebut akan kita pelajari pada beberapa materi ke depan.

### Gathering Data

Proses pengumpulan data merupakan salah satu tahapan paling menantang dalam proyek analisis data. Bergantung pada proyek analisis data yang dikerjakan, data yang Anda butuhkan mungkin terdapat dalam berbagai sumber dan memiliki format yang berbeda-beda.

Salah satu skill yang harus dimiliki oleh seorang praktisi data ialah mengetahui sumber data yang tepat untuk mengumpulkan data yang dibutuhkan. Berikut merupakan beberapa sumber data yang bersifat publik dan sering digunakan oleh para praktisi data.

* **Kaggle**  
  Kaggle merupakan platform andalan bagi praktisi data untuk mencari dataset. Selain menyediakan dataset, kaggle juga menyediakan berbagai kompetisi dan challenge terkait data science dan machine learning. Kaggle juga memungkinkan para praktisi data dan machine learning saling berbagai kode dan pengetahuan terkait data serta machine learning.
* **UCI Machine Learning Repository**  
  UCI Machine Learning Repository merupakan sebuah repositori yang menampung berbagai dataset yang bersifat publik.
* **Google Dataset Search**  
  Google Dataset Search merupakan sebuah search engine yang disediakan oleh Google. Ia dibuat untuk mempermudah para praktisi data dan juga researcher dalam mencari dataset yang bersifat publik.
* **Satu Data Indonesia**  
  Tahukah Anda bahwa pemerintah Indonesia sebetulnya telah menyediakan sebuah platform bernama Satu Data Indonesia atau sering disingkat SDI.

Selain menggunakan berbagai sumber dataset publik tersebut, para praktisi data juga seringkali menggunakan data internal yang bersifat privat. Data seperti ini umumnya disimpan dalam sebuah sistem pengolahan database.

Pandas menyediakan beberapa function yang dapat digunakan untuk membaca atau mengakses data menjadi sebuah DataFrame. Saat ini, Pandas mendukung pembacaan atau pengaksesan data dari berbagai format berkas data dan sumber data seperti pada contoh di bawah ini.

* **Format berkas CSV**  
  Berkas CSV (Comma Separated Values) merupakan format berkas data tabel yang paling sering digunakan dan telah menjadi standar dalam industri. Pandas menyediakan sebuah function **read\_csv()**untuk membaca berkas CSV.
* **Format berkas XLSX atau XLS**  
  Berkas XLSX atau XLS merupakan format berkas spreadsheet yang dibuat menggunakan aplikasi Microsoft Excel. Pandas juga menyediakan function **read\_excel()** untuk membaca berkas data dengan format XLSX atau XLS.
* **Format berkas JSON**  
  Berkas JSON (JavaScript Object Notation) merupakan format berkas data yang memiliki struktur data yang mirip seperti data structure dictionary dalam Python yang terdiri dari pasangan keys dan values. Pandas juga menyediakan function **read\_json()** untuk membaca berkas data berformat JSON.
* **Format berkas HTML**  
  HTML atau dikenal juga sebagai HyperText Markup Language merupakan sebuah markup language standar yang digunakan untuk merancang tampilan sebuah dokumen/halaman di web browser. Untuk membaca berkas ini, pandas menyediakan function **read\_html()**.
* **Format berkas XML**  
  Format data selanjutnya yang akan kita bahas ialah XML yang merupakan singkatan dari Extensible Markup Language. Ia sering digunakan untuk merepresentasikan berbagai struktur informasi, seperti dokumen, data, konfigurasi, dll. Pandas menyediakan function **read\_xml()** untuk membaca format data ini.
* **Akses data dari SQL database**  
  Selain membaca data dari berbagai format, library pandas juga memungkinkan kita untuk mengakses data langsung dari sebuah database, seperti PostgreSQL, MySQL, dll. Tentunya untuk mengakses database tersebut kita membutuhkan library pendukung yaitu SQLAlchemy.  
    
  Untuk berinteraksi dengan database, pandas menyediakan tiga function seperti berikut.
  + **read\_sql\_table()** untuk membaca SQL database table dan mempresentasikannya ke dalam bentuk pandas DataFrame.
  + **read\_sql\_query()** untuk membaca SQL query dan mempresentasikannya ke dalam bentuk pandas DataFrame.
  + **read\_sql()** untuk membaca SQL query atau table dan mempresentasikannya ke dalam bentuk pandas DataFrame.

Seperti yang telah kita bahas sebelumnya, untuk menyelesaikan sebuah permasalahan bisnis, terkadang kita perlu menggunakan lebih dari satu tabel data. Tentunya untuk melakukan analisis kita perlu menggabungkan data tersebut. Salah satu teknik penggabungan data yang paling sering digunakan ialah merge atau join.

Berdasarkan cara penggabungannya, proses merge atau join dapat dibagi menjadi empat jenis yaitu seperti berikut.

* **Inner**  
  Inner join merupakan proses join yang hanya mengambil nilai yang bersesuaian di kedua tabel.
* **Left**  
  Left join merupakan proses join yang akan mengambil semua nilai dari tabel kiri beserta nilai yang bersesuaian dari tabel kanan.
* **Right**  
  Right join merupakan proses join yang akan mengambil semua nilai dari tabel kanan beserta nilai yang bersesuaian dari tabel kiri. Ia merupakan kebalikan dari left join.
* **Outer**  
  Outer join atau sering juga disebut full outer join merupakan proses join yang akan mengambil semua nilai dari kedua tabel. Ia merupakan gabungan dari left dan right join.

Sebagai tool andalan dalam pengolahan dan analisis data, pandas menyediakan sebuah function bernama **merge()**. Ia dapat digunakan untuk menggabungkan dua buah DataFrame.

### Assessing Data

Pemeriksaan data ini dilakukan dengan menjalankan proses assessing data. Ia merupakan proses yang bertujuan untuk mengidentifikasi masalah yang terdapat dalam data dan memastikan data tersebut berkualitas.

Seperti yang telah kita bahas sebelumnya, kebersihan dan kualitas dari sebuah data merupakan tantangan utama dalam proyek analisis data di industri. Data yang kotor umumnya memiliki masalah dalam kontennya. Berikut beberapa masalah yang umum dijumpai dalam sebuah data.

* **Missing value**  
  Masalah ini muncul karena adanya nilai yang hilang dari sebuah data dan biasanya direpresentasikan sebagai nilai **NaN** dalam library pandas. Hal ini biasanya terjadi karena adanya human error, masalah privasi, proses merging/join, dll.  
    
  Library pandas menyediakan sebuah method bernama **isnull()** atau**isna()** untuk mengidentifikasi missing value dalam sebuah DataFrame. Keduanya sering dipadukan dengan method **sum()** untuk menghitung total missing value pada setiap kolom dalam sebuah DataFrame.
* **Invalid value**  
  Masalah ini muncul ketika terdapat beberapa nilai yang tidak masuk akal, tidak sesuai dengan ketentuan, dan background knowledge dari data tersebut.  
    
  Untuk mendeteksi masalah seperti ini, kita membutuhkan teknik sedikit advance. Salah satu teknik yang paling sering digunakan ialah teknik filtering data menggunakan regex.
* **Duplicate data**  
  Duplicate data merupakan masalah lain yang umum dijumpai di industri. Ia terjadi ketika terdapat sebuah observasi (semua nilai dalam satu unit baris) yang memiliki nilai yang sama persis pada setiap kolomnya. Pandas menyediakan sebuah method **duplicated()** untuk mengidentifikasi apakah terdapat duplikasi pada sebuah DataFrame.
* **Inaccurate value**  
  Inaccurate value merupakan masalah yang muncul ketika nilai dalam sebuah data tidak sesuai dengan hasil observasi. Masalah ini umumnya muncul karena adanya human error atau sistem error.
* **Inconsistent value**  
  Inconsistent value adalah masalah yang muncul ketika sebuah data memiliki nilai yang tidak konsisten baik dari segi satuan maupun ketentuan penilaian. Inkonsistensi ini umumnya muncul karena adanya perbedaan standar dalam proses pengumpulan nilai.
* **Outlier**  
  Outlier atau dalam bahasa indonesia disebut pencilan merupakan titik data yang berada sangat jauh dari titik data yang lain dalam sebuah dataset. Nilai yang sangat jauh ini tentunya akan berdampak terhadap beberapa parameter statistik yang digunakan untuk menganalisis data, seperti nilai mean dan standard deviation.  
    
  Terdapat beberapa metode yang dapat digunakan untuk mengidentifikasi outlier dalam sebuah dataset. Metode yang paling sering digunakan ialah IQR method.  
    
  IQR method merupakan metode penentuan outlier berdasarkan nilai interquartile range (IQR). Ia mengidentifikasi outlier dengan cara membuat nilai cut-off sebagai faktor k (Umumnya kita menggunakan nilai 1.5 s/d 3) dari nilai IQR. Cut-off tersebut selanjutnya akan digunakan untuk menghitung nilai ambang batas (boundary values).  
    
  Metode lain yang bisa digunakan ialah box plot. Ia merupakan bentuk visual untuk merepresentasikan nilai IQR beserta ambang batas bawah dan atas dari sebuah data. Hal ini tentunya akan membantu kita mengidentifikasi outlier secara lebih mudah yaitu melalui bentuk visual.

### Cleaning Data

Data yang kotor juga harus kita bersihkan untuk memastikan ia tidak mempengaruhi hasil analisis yang akan kita lakukan nantinya.

Secara umum, proses pembersihan data dapat dibagi ke dalam tiga tahapan, yaitu define, code, dan test.

* **Define:** pada tahap ini, kita akan membuat rancangan tahapan serta metode pembersihan data berdasarkan masalah yang ditemukan dalam proses assessing data.
* **Code:** setelah membuat rancangan pembersihan data, tahap selanjutnya ialah mengonversi hal tersebut menjadi sebuah kode program yang dapat dijalankan.
* **Test:** setelah menjalankan kode program untuk membersihkan data, kita perlu memeriksa kembali data yang telah dibersihkan tersebut.

#### Teknik untuk Mengatasi Missing Value

Secara garis besar, terdapat tiga metode dalam mengatasi missing value antara lain seperti berikut.

* **Dropping**  
  Metode pertama dan yang paling mudah dalam mengatasi missing value adalah dropping. Pada metode ini, kita akan menghapus seluruh baris atau kolom yang memiliki missing value. Untuk melakukannya, kita bisa menggunakan salah satu method **dropna()** yang disediakan oleh library pandas  
    
  Metode dropping ini memang terkesan mudah, tetapi perlu Anda ketahui bahwa ada banyak sekali pertimbangan yang perlu diperhatikan sebelum menggunakan metode ini. Salah satu pertimbangan penting ialah penerapan metode ini dapat menyebabkan kita kehilangan banyak informasi.
* **Imputation**  
  Metode ini bekerja dengan cara mengisi (fill) missing value dengan nilai tertentu. Hal ini tentunya akan mencegah hilangnya informasi akibat missing value.  
    
  Pada data kontinu, kita bisa menggunakan nilai mean, median, atau mode sebagai pengganti missing value. Jika bekerja menggunakan data kategoris, kita dapat mengisi missing value dengan kategori yang paling sering muncul. Namun, perlu diingat bahwa pemilihan nilai pengganti ini harus didukung oleh background knowledge dari data tersebut.  
    
  Sebagai library andalan kita, pandas telah menyediakan sebuah method bernama **fillna()** untuk mengganti missing value dengan nilai tertentu.  
    
  Metode ini masih memiliki banyak kekurangan salah satunya ialah dapat mempengaruhi variance atau sebaran dari sebuah data. Selain itu, metode ini juga masih belum cukup baik untuk diterapkan pada data time series.
* **Interpolation**  
  Metode penanganan missing value terakhir yang akan kita bahas ialah interpolation (interpolasi). Sederhananya, interpolasi merupakan salah satu pendekatan numerik yang digunakan untuk menghitung titik data baru berdasarkan range data yang sudah ada. Perhitungan ini menggunakan sebuah persamaan garis linear ataupun polynomial. Library pandas juga menyediakan method **interpolate()** yang bisa kita gunakan untuk menerapkan metode interpolasi dalam mengatasi missing value.

#### Teknik untuk Mengatasi Outlier

Pada dasarnya, terdapat dua metode yang umum digunakan untuk mengatasi outlier yaitu drop dan imputation.

* **Drop**  
  Metode pertama yang paling mudah ialah men-drop atau menghapus seluruh baris yang mengandung outlier. Metode ini mampu mencegah outlier mempengaruhi hasil analisis yang kita buat.
* **Imputation**  
  Metode lain yang bisa Anda gunakan untuk menangani outlier ialah imputation. Konsepnya mirip seperti sebelumnya yaitu mengganti outlier dengan nilai tertentu. Nilai yang bisa kita gunakan ialah mean, media, dan mode. Selain itu, tidak jarang juga kita mengganti outlier dengan boundary value.  
    
  Untuk menerapkan metode ini, kita bisa menggunakan method **mask()** yang disediakan oleh library pandas. Method tersebut menerima parameter **cond** sebagai kondisi untuk memfilter nilai outlier.

#### Teknik untuk Mengatasi Duplicate Data

Ketika menemukan duplikasi pada data, tentunya kita harus menghilangkan atau menghapus duplikasi tersebut. Library pandas telah menyediakan sebuah method **drop\_duplicates()** untuk menghilangkan duplikasi dalam sebuah DataFrame.

**Rangkuman Exploratory Data Analysis**

**Pengenalan Exploratory Data Analysis**

Exploratory Data Analysis (EDA) merupakan tahap eksplorasi data yang telah dibersihkan guna memperoleh*insight*dan menjawab pertanyaan analisis. Pada prosesnya, kita akan sering menggunakan berbagai teknik dan parameter dalam *descriptive statistics* yang bertujuan untuk menemukan pola, hubungan, serta membangun intuisi terkait data yang diolah. Selain itu, tidak jarang kita juga menggunakan visualisasi data untuk menemukan pola dan memvalidasi parameter descriptive statistics yang diperoleh.

Exploratory analysis dan explanatory analysis merupakan dua istilah sering dijumpai dalam proyek analisis data.

Sederhana, exploratory analysis merupakan proses analisis data yang bertujuan untuk mengeksplorasi dan mengenal sebuah data. Pada prosesnya sering kali mulai dengan mendefinisikan berbagai pertanyaan atau hanya menggali data guna menemukan beberapa insight yang menarik dari sebuah data.

Di lain sisi, explanatory analysis merupakan proses analisis data yang bertujuan untuk membagikan beberapa insight yang menarik dari sebuah data. Insight inilah yang akan digunakan untuk menjawab permasalahan bisnis yang dihadapi.

Jika disimpulkan, explanatory analysis merupakan proses penyampaian temuan menarik dari proses exploratory analysis. Proses penyampaian ini tentunya harus diikuti dengan visualisasi data yang baik dan efektif.

**Mendefinisikan Pertanyaan Untuk Explorasi Data**

Sebagai calon praktisi data masa depan, tentunya Anda telah mengetahui bahwa objektif dari sebuah proyek analisis data ialah menjawab tantangan atau permasalahan bisnis. Untuk memastikan pemahaman terhadap sebuah permasalahan bisnis, kita perlu mengurai masalah tersebut ke dalam beberapa pertanyaan analisis.

Problem solving merupakan salah satu keterampilan penting yang harus dimiliki seorang praktisi data. Untuk menyelesaikan sebuah problem, kita harus memahami terlebih dahulu permasalahan yang ingin diselesaikan. Berikut merupakan beberapa kategori problem yang sering dijumpai oleh para praktisi data.

* **Membuat prediksi**  
  Membuat prediksi merupakan salah satu problem yang paling sering dijumpai oleh praktisi data di industri. Jenis problem ini, mengharuskan kita untuk membuat sebuah prediksi berdasarkan hasil analisis dari data sebelumnya.
* **Mengategorikan berbagai hal**  
  Problem lain yang umum dijumpai di industri ialah kebutuhan untuk mengategorikan berbagai hal. Pada problem ini, kita harus menetapkan sebuah informasi ke dalam berbagai grup atau kluster berdasarkan kesamaan yang ada.
* **Menemukan sesuatu yang tidak biasa**  
  Kategori problem selanjutnya ialah mengidentifikasi atau menentukan sesuatu yang tidak biasa. Problem ini, umumnya muncul ketika terdapat sebuah kejadian aneh yang perlu dianalisis lebih lanjut.
* **Mengidentifikasi tema**  
  Permasalahan lain yang mungkin akan Anda temui ialah mengidentifikasi tema. Problem ini sering dijumpai dalam bidang UI/UX. Seorang UI/UX *designer* harus mengidentifikasi tema yang paling tepat dalam merepresentasikan *anonymous survey data*dari para pengguna. Hal ini bertujuan untuk meningkatkan pengalam pengguna ketika menggunakan sebuah produk.
* **Menemukan hubungan**  
  Kategori berikutnya ialah menemukan hubungan atau *connection*. Problem ini mengharuskan kita untuk mengidentifikasi dan menganalisis hubungan dari berbagai data untuk mengatasi suatu permasalahan.
* **Menemukan pola**  
  *Last but not least*, kategori problem yang mungkin Anda temui ialah menemukan pola. Jenis problem ini, membutuhkan analisis mendalam terhadap sebuah data guna memahami pola yang terdapat di dalamnya.

Jika data merupakan building block dalam proyek analisis data, kita dapat menganggap pertanyaan analisis sebagai kompas yang selalu mengarahkan untuk menuju goals dan objektif yang ingin dicapai. Pada proses pembuatan pertanyaan analisis, kita perlu menghindari beberapa poin berikut.

* Hindari membuat pertanyaan yang mengarahkan kita untuk menjawab dengan jawaban tertentu.
* Hindari membuat pertanyaan yang hanya dapat dijawab dengan “ya” atau “tidak.”
* Hindari membuat pertanyaan yang tidak memiliki konteks.

Untuk menghasilkan pertanyaan analisis yang efektif, terdapat beberapa poin yang harus Anda perhatikan seperti berikut.

* **Specific**  
  Pertanyaan yang efektif haruslah bersifat spesifik, artinya sebuah pertanyaan analisis harus fokus pada sebuah topik tertentu. Pertanyaan yang spesifik dapat membantu kita dalam mengumpulkan informasi yang relevan dengan tujuan dari proyek analisis data.
* **Measurable**  
  Selain bersifat spesifik, pertanyaan yang efektif haruslah bisa diukur atau *measurable*. Hal ini tentunya akan mempermudah kita dalam memahami dan menarik insight dari sebuah data.
* **Action-oriented**  
  Pertanyaan yang bersifat *action-oriented* berarti pertanyaan tersebut mampu memberikan informasi yang dapat membantu kita dalam membuat sebuah action-plan.
* **Relevant**  
  Pertanyaan yang relevan berarti pertanyaan tersebut berbobot dan memiliki arti untuk menyelesaikan sebuah permasalahan.
* **Time-bound**  
  Terakhir, pertanyaan yang efektif haruslah bersifat time-bound yang berarti pertanyaan tersebut haruslah memiliki relevansi terhadap waktu saat ini.

**Mengeksplorasi Parameter Statistik dari Data**

Method **describe()** memungkinkan kita untuk melihat rangkuman parameter statistik (mean, median, dll.) dari kolom numerik pada sebuah DataFrame. Jika ingin melihat lebih banyak parameter statistik serta menyertakan kolom non-numerik, Anda bisa menggunakan parameter **include=”all”**.

Selain method **describe()**, pandas juga menyediakan method lain seperti **hist()** untuk membuat plot histogram dari kolom numerik pada sebuah DataFrame. Method ini dapat digunakan untuk memeriksa distribusi data secara cepat.

Jika ingin memeriksa korelasi antar data numerik pada sebuah DataFrame, Anda dapat menggunakan method **corr()** atau **cov()**.

Itulah beberapa method yang sering digunakan untuk mengeksplorasi parameter statistik dari sebuah DataFrame. Sebenarnya masih banyak method lain yang bisa Anda gunakan untuk memeriksa parameter statistik lain. Silahkan kunjungi dokumentasi berikut untuk melihat lebih detail terkait hal tersebut: [dokumentasi pandas.DataFrame](https://pandas.pydata.org/docs/reference/frame.html).

**Mengelompokkan Data**

Mungkin beberapa dari Anda cukup familier dengan konsep *pivot table* dalam aplikasi pengolahan data seperti Microsoft Excel atau Google Sheets. Ia merupakan tool yang dapat digunakan untuk mengelompokkan data berdasarkan parameter tertentu.

Pada proses analisis data, kita bisa menggunakan konsep pivot table ini untuk menghitung parameter statistik dari kelompok data tertentu. Selain itu, ia juga dapat digunakan untuk kebutuhan visualisasi data dan pembuatan *report*.

Untuk memahami proses pembuatan pivot table, Anda perlu mengenal dua istilah di bawah ini telebih dahulu.

* Key: kolom yang berisi data kategorik dan digunakan untuk mengelompokkan data.
* Value: kolom yang berisi data kuantitatif dan akan dihitung nilai parameter statistiknya

Proses pembuatan pivot table, diawali dengan melakukan split atau proses pengelompokkan pasangan key-values. Tahap berikutnya adalah proses apply atau perhitungan parameter dari setiap kelompok pasangan key-value. Tahap terakhir ialah menggabungkan (*combine*) seluruh key dengan nilai parameternya ke dalam satu tabel. *Tara*…, jadilah sebuah pivot table yang akan membantu kita dalam memahami sebuah data.

Library pandas telah menyediakan sebuah method bernama groupby() yang dapat membantu kita dalam membuat pivot table secara lebih efisien dan scalable. Method ini akan menerima inputan beberapa parameter salah satunya ialah “by”. Parameter tersebut digunakan untuk menentukan key dari pivot table tersebut. Selain itu, untuk menggunakan method ini, kita perlu memadukannya dengan berbagai method untuk menghitung parameter statistik, seperti mean(), mode(), dll.

Untuk membuat pivot table dengan lebih dari satu parameter, kita bisa menggunakan method agg(). Ia memungkinkan kita untuk menentukan beberapa parameter yang ingin digunakan dalam pivot table.

Nah, itulah segenap pembahasan terkait pengelompokkan data atau pembuatan pivot table menggunakan method groupby().  Jika Anda ingin membaca lebih detail terkait method tersebut, silakan kunjungi tautan berikut: [GroupBy documentation](https://pandas.pydata.org/pandas-docs/stable/reference/groupby.html).

## Rangkuman Data Visualization

### Pengenalan Data Visualization

Pada siklus tersebut, data visualization merupakan tahapan yang harus kita lakukan sebelum membuat kesimpulan dan mengomunikasikan (draw conclusion & communicate) hasil dari proses analisis yang telah dilakukan. Hal ini dilakukan untuk mempermudah kita dalam memahami data dan membuat sebuah kesimpulan yang andal. Selain itu, visualisasi data juga dapat membantu kita dalam menyampaikan cerita atau temuan dari hasil analisis data kepada orang lain atau stakeholder.

Sederhananya, visualisasi data merupakan cara kita dalam menyajikan data dalam bentuk visual. Hal ini dilakukan untuk mempermudah kita dan orang lain dalam memahami data tersebut. Selain itu, visualisasi data yang baik juga akan sangat membantu kita dalam menyampaikan story dan pesan dari sebuah data.

### Prinsip-Prinsip dalam Visualisasi Data

Ketika berbicara tentang visualisasi data, kita tidak hanya membahas cara menyajikan data ke dalam bentuk visual, melainkan juga harus memahami berbagai prinsip dasar dalam membuat visualisasi data yang baik dan benar.

Umumnya visualisasi data yang buruk memuat salah satu dari tiga masalah berikut.

* **Misleading**  
  Masalah pertama dalam visualisasi data yang buruk ialah misleading information yang mampu mengakibatkan kesalahan dalam pengambilan kesimpulan dari sebuah data.
* **Hides**  
  Selain misleadning information, masalah lain yang umum dijumpai dalam visualisasi data yang buruk ialah menyembunyikan informasi tertentu.
* **Distracts**  
  Masalah lain yang mungkin terjadi ialah distraksi. Visualisasi data yang buruk sering kali menyertakan komponen visual yang seharusnya tidak dibutuhkan dan malah mengganggu proses pengambilan kesimpulan dari sebuah visualisasi data.

Jika ditelisik secara mendasar, visualisasi data merupakan proses dalam mengubah data ke dalam bentuk visual menggunakan berbagai elemen visual. Berikut merupakan beberapa elemen visual yang umum digunakan untuk membuat visualisasi data.

* **Position**: elemen ini akan membantu kita merepresentasikan titik data menggunakan sumbu tertentu (seperti sumbu X, Y, dan Z) sebagai acuan.
* **Size**: ukuran (panjang atau lebar) merupakan elemen visual yang umumnya kita gunakan untuk membedakan serta membandingkan nilai dari kategori atau titik data tertentu.
* **Shape**: bentuk merupakan salah satu elemen visual yang dapat digunakan untuk membedakan kategori atau titik data tertentu.
* **Color**: selain bentuk, warna juga merupakan pilihan elemen visual lain yang dapat digunakan untuk membedakan kategori atau titik data tertentu. Ketika menggunakan elemen ini, kita harus ingat bahwa tidak semua orang memiliki kemampuan untuk membedakan warna dengan baik.
* **Texture**: penambahan tekstur atau pola tertentu bisa menjadi alternatif lain dalam membedakan kategori atau titik data tertentu.
* **Angle**: pada beberapa pilihan bentuk visualisasi data, sudut merupakan salah satu elemen visual yang digunakan untuk merepresentasikan nilai dari suatu data.

Untuk memastikan penggunaan elemen visual secara efektif, Edward Tufte telah mengenalkan sebuah konsep bernama **Data Ink Ratio**pada bukunya yang berjudul “The Visual Display of Quantitative Data”.

Data ink ratio merupakan sebuah konsep parameter yang digunakan sebagai penentu efektivitas penggunaan visual elemen dalam sebuah visualisasi data. Ia didefinisikan sebagai perbandingan antara tinta (bisa diartikan sebagai elemen visual) yang digunakan untuk mendeskripsikan data dan total tinta yang digunakan dalam satu visualisasi data.

Ketika membuat sebuah visualisasi data, kita tidak hanya memikirkan terkait faktor estetika (seni) saja. Melainkan, kita juga harus memastikan visualisasi data yang kita buat mampu menyampaikan berbagai informasi penting dari sebuah data tanpa menyembunyikan fakta, mengarahkan orang lain pada asumsi tertentu, serta menghindari misinterpretasi.

Pada bukunya yang berjudul The Visual Display of Quantitative Data, Edward Tufte juga mengenalkan beberapa prinsip guna menjaga integritas dalam membuat visualisasi data. Berikut beberapa prinsip tersebut.

* Representasi angka pada sebuah grafik harus sebanding dengan kuantitas sebenarnya. Hal ini untuk menghindari terjadinya misinterpretasi dari sebuah visualisasi data.
* Pelabelan yang jelas, terperinci, dan menyeluruh merupakan kunci utama dalam menghindari ambiguitas dalam sebuah visualisasi data.
* Visualisasi data yang baik harus menunjukkan variasi pada data bukan variasi desain.
* Untuk menghindari asumsi, kita harus menggunakan unit yang umum digunakan (sesuai standar atau kesepakatan) dalam merepresentasikan suatu data tertentu.
* Jumlah variabel (feature/kolom) yang digambarkan tidak boleh melebihi jumlah dimensi dalam data. Sebagai contoh, untuk menggambarkan hubungan variabel X dan Y, kita disarankan untuk menggunakan plot dua dimensi ketimbang menggunakan plot tiga dimensi.
* Sebuah grafik tidak boleh menampilkan data yang di luar konteks.

Nah, itu dia keenam prinsip yang harus kita ikuti untuk menjaga integritas dan menghindari misinterpretasi dalam sebuah visualisasi data. Sebenarnya pada buku yang sama, Edward Tufte juga mengenalkan sebuah parameter bernama **Lie Factor** untuk mengukur kesalahan representasi dalam sebuah visualisasi data. Parameter ini didefinisikan sebagai rasio perbandingan ukuran yang ditampilkan pada grafik dan ukuran yang sebenarnya ada dalam data. Rasio ini tentunya akan memberikan kita gambaran sejauh mana visualisasi mendistorsi nilai dari sebuah data.

### Univariate Visualization

Seperti yang telah kita bahas sebelumnya, visualisasi data merupakan teknik yang digunakan dalam merepresentasikan atau menyajikan data ke dalam bentuk visual yang lebih mudah dipahami. Nah, berdasarkan jumlah variabel (feature/kolom) yang ingin disajikan, visualisasi data dapat dibagi menjadi tiga kategori.

* **Univariate visualization**  
  Univariate visualization merupakan bentuk visualisasi data yang hanya merepresentasikan informasi yang terdapat pada satu variabel. Jenis visualisasi ini umumnya digunakan untuk memberikan kita gambaran terkait distribusi sebuah variabel dalam suatu dataset.
* **Bivariate visualization**  
  Bivariate visualization merupakan bentuk visualisasi data untuk menyajikan informasi yang terdapat dalam dua variabel. Visualisasi jenis ini dapat digunakan untuk menganalisis hubungan antar dua variabel dalam sebuah dataset.
* **Multivariate visualization**  
  Last but not least, multivariate visualization merupakan jenis visualisasi data untuk menggambarkan informasi yang terdapat dalam lebih dari dua variabel. Jenis visualisasi ini digunakan untuk merepresentasikan hubungan dan pola yang terdapat dalam multidimensional data.

#### Bar Chart

Bentuk visualisasi data pertama yang akan kita bahas ialah bar chart atau dikenal juga dengan sebutan diagram batang. Ia merupakan pilihan visualisasi data yang digunakan untuk menggambarkan distribusi dari suatu data kategoris (categorical data).

Pada bar chart, setiap kategori akan direpresentasikan sebagai sebuah batang. Tinggi dari sebuah batang menunjukkan frekuensi atau jumlah titik data dari kategori tersebut.

Library matplotlib telah menyediakan sebuah function bernama **pyplot.bar()** untuk membuat sebuah bar chart. Untuk menggunakan function tersebut, kita perlu mendefinisikan parameter **x** dan **height** dari bar chart yang ingin dibuat.

Selain menggunakan library matplotlib, kita juga bisa membuat bar chart menggunakan library seaborn. Untuk membuat bar chart menggunakan library seaborn, kita bisa menggunakan function **barplot()**. Function ini akan menyediakan beberapa parameter penting seperti berikut.

* **data**: menampung DataFrame yang akan digunakan.
* **x, y**: menampung nama kolom atau data yang divisualisasikan.
* **orient**: orientasi dari bar chart yang akan digunakan (“v” atau “h”).
* **color**: mendefinisikan warna yang akan digunakan.

#### Pie Chart

Pie chart merupakan bentuk visualisasi data yang dapat digunakan untuk menggambarkan perbandingan frekuensi tiap kategori pada suatu data kategoris. Pada pie chart, setiap kategori digambarkan oleh irisan dalam sebuah lingkaran dan frekuensi digambarkan sebagai luasan (area) dari irisan tersebut.

Terdapat variasi lain dari pie chart, yaitu donut plot. Sebenarnya, ia merupakan pie chart yang memiliki lubang pada bagian tengahnya sehingga membentuk donut. Selain itu, tidak ada perbedaan mendasar dalam penggunaan antara pie chart dan donut plot.

Untuk membuat pie chart, kita bisa memanfaatkan library andalan, yaitu matplotlib. Ia telah menyediakan sebuah function bernama **pie()** untuk menghasilkan grafik pie chart. Function tersebut memiliki beberapa parameter penting yaitu seperti berikut.

* **x**: menampung data yang akan divisualisasikan.
* **explode**: menampung array atau list yang mengatur posisi tiap irisan lingkaran.
* **labels**: sekumpulan string yang digunakan untuk memberi label pada tiap irisan lingkaran.
* **colors**: sekumpulan warna yang akan digunakan pada tiap irisan lingkaran.
* **autopct**: string yang digunakan untuk memberi numerik label pada tiap irisan lingkaran.

#### Histogram

Jika dilihat sekilas, bentuk grafik histogram mirip dengan bentuk bar chart perbedaannya hanya terdapat pada sumbu X. Pada grafik histogram, sumbu X digunakan untuk menampung range nilai dari data kuantitatif yang dikenal dengan istilah bins.

Pada grafik histogram, ukuran bins akan sangat berpengaruh terhadap cara kita dalam merepresentasikan data kuantitatif. Jika menggunakan ukuran bins yang terlalu besar, kita akan kehilangan cukup banyak informasi dari data kuantitatif yang ingin divisualisasikan. Sebaliknya, apabila ukuran bins yang digunakan terlalu kecil, hal ini akan menghasilkan banyak noise yang mendistraksi distribusi data yang ingin kita amati.

Untuk membuat grafik histogram, kita bisa menggunakan function **hist()**yang disediakan oleh library matplotlib. Function ini menerima beberapa parameter seperti berikut.

* **x**: menampung data yang akan divisualisasikan.
* **bins**: menampung jumlah bins (sebanding dengan ukurannya) yang akan digunakan untuk membuat grafik histogram.

Selain menggunakan library matplotlib, kita juga bisa membuat grafik histogram dengan library seaborn. Library ini menyediakan function **histplot()** untuk membuat grafik histogram.

#### Box Plot

Bentuk visualisasi data lain yang bisa digunakan untuk mengidentifikasi distribusi pada data kuantitatif ialah box plot. Ia merupakan bentuk visual untuk merepresentasikan nilai IQR beserta ambang batas bawah dan atas dari sebuah data. Box plot ini juga bisa digunakan untuk mengidentifikasi outlier atau pencilan yang ada pada data kuantitatif. Kita bisa menggunakan function**boxplot()** yang disediakan oleh library seaborn membuat plot jenis ini.

### Bivariate dan Multivariate Visualization

Sederhananya, bivariate visualization merupakan bentuk visualisasi data untuk menyajikan informasi dari dua variabel dalam suatu dataset. Di lain sisi, multivariate visualization merupakan jenis visualisasi data untuk menggambarkan informasi yang terdapat dalam lebih dari dua variabel. Keduanya sering digunakan untuk mendeskripsikan hubungan antar variabel dalam suatu dataset.

#### Scatter Plot

Scatter plot merupakan bentuk visualisasi data yang digunakan untuk melihat hubungan antara dua atau lebih variabel data kuantitatif. Pada scatter plot sumbu X, Y, atau Z digunakan untuk menampung nilai dari setiap variabel yang akan divisualkan. Titik pada scatter plot merupakan titik temu data dari setiap sumbu yang digunakan.

Untuk membuat scatter plot, kita bisa menggunakan library matplotlib. Ia telah menyediakan sebuah function bernama **scatter()** untuk membantu kita membuat scatter plot secara lebih mudah.

Selain menggunakan library matplotlib, kita juga bisa menggunakan function **scatterplot()** yang disediakan oleh library seaborn untuk membuat scatter plot.

Untuk mempermudah dalam melihat korelasi atau hubungan antar variabel, kita bisa menggunakan function **regplot()** yang disediakan oleh library seaborn. Function tersebut memadukan scatter plot dan regression function (metode statistik untuk memperkirakan korelasi antar independent dan dependent variable) untuk melihat tren serta korelasi antar variabel.

#### Line Chart

Line chart merupakan bentuk grafik yang umum digunakan untuk menggambarkan tren nilai dari suatu variabel terhadap variabel lain. Kita bisa membuat line chart menggunakan function **plot()** yang disediakan oleh matplotlib.

Bentuk visualisasi data ini umumnya digunakan untuk melihat trend dari data berbentuk time series (data yang direkam dalam interval waktu yang konsisten).

#### Clustered Bar Chart

Clustered bar chart merupakan bentuk modifikasi dari bar chart yang sebelumnya kita kenal dengan menambahkan variabel kategoris lain. Modifikasi ini memungkinkan untuk melihat distribusi serta hubungan antar dua atau lebih variabel kategoris.

Untuk membuat clustered bar chart, kita bisa menggunakan parameter **hue** pada function **barplot()** yang disediakan oleh library seaborn.

#### Menggunakan Komponen Visual Tambahan

Selain menggunakan scatter plot, line chart, dan clustered bar chart, kita juga bisa memodifikasi bentuk visualisasi data dengan menambahkan komponen visual tambahan seperti color, shape, serta texture untuk mendeskripsikan lebih banyak variabel.

Sebagai contoh, Anda bisa membuat lebih dari satu boxplot untuk menggambarkan distribusi nilai numerik dari beberapa variabel. Hal ini bisa Anda gunakan dengan memasukkan DataFrame yang akan divisualisasikan pada parameter **data**dalam function **boxplot()**. Selain itu, Anda juga bisa menggunakan parameter **palette** untuk mengatur palet warna yang akan digunakan.

### Explanatory Analysis

Explanatory analysis merupakan proses analisis data yang bertujuan untuk membagikan beberapa insight yang menarik dari sebuah data. Insight yang diperoleh dari proses explanatory analysis akan digunakan untuk menjawab permasalahan bisnis yang dihadapi. Insight tersebut juga akan dibagikan kepada orang lain (stakeholder) sehingga perlu disertakan dengan visualisasi data yang baik

Pada proses explanatory analysis, visualisasi data digunakan sebagai media untuk menyampaikan pesan atau story dari sebuah data. Oleh karena itu, kita harus menyampaikan informasi terkait konteks dari sebuah visualisasi data. Hal ini bisa dicapai dengan menambahkan beberapa komponen berikut.

* **Title**  
  Title atau judul merupakan salah satu komponen penting guna menyampaikan konteks pada sebuah visualisasi data. Kita bisa menambahkan judul pada visualisasi data menggunakan function **title()** yang disediakan oleh matplotlib.
* **Label axes**  
  Selain judul visualisasi, kita juga perlu menambahkan label atau keterangan pada setiap sumbu yang terdapat dalam visualisasi data. Hal ini dilakukan untuk memberikan konteks terkait titik data yang direpresentasikan. Kita bisa membuat label axes menggunakan function **xlabel()** dan **ylabel()** yang disediakan oleh library matplotlib.
* **Legend**  
  Pastikan Anda menambahkan legend memberikan keterangan untuk variabel yang tidak digambarkan pada sumbu. Keterangan yang digunakan akan bergantung pada cara kita dalam merepresentasikan variabel tambahan tersebut. Sebagai contoh, kita bisa menggunakan **legend()** untuk memberikan keterangan pada variabel kategoris tambahan. Namun, apabila variabel tambahan tersebut berupa nilai numerik yang direpresentasikan menggunakan range warna, kita bisa memanfaatkan function **colorbar()** untuk memberikan keterangan pada variabel tersebut.

Selain informasi konteks, pastikan audiens fokus pada informasi yang ingin kita sampaikan. Oleh karena itu, sebagai seorang praktisi data yang andal, kita harus mampu mengarahkan perhatian audiens. Untuk melakukan hal ini, kita bisa menggunakan komponen visual tertentu seperti color, shape, dan texture.

Sebagai contoh, kita bisa menggunakan warna yang kontras untuk mengarahkan perhatian audiens pada bagian yang ingin di-highlight.

Namun, perlu diingat bahwa ketika menggunakan teknik ini kita harus menghindari penggunaan warna yang menyakitkan mata audiens atau warna yang memiliki arti berlawanan dengan pesan yang ingin disampaikan. Selain itu, kita juga memperhatikan konsistensi dalam pemilihan warna dalam seluruh visualisasi data yang kita buat.

Pada proses pembuatan explanatory analysis, terkadang sulit untuk menilai seberapa efektif visualisasi data yang kita buat. Oleh karena itu, kita membutuhkan feedback serta sudut pandang orang lain guna memastikan bahwa visualisasi data yang kita buat sudah cukup efektif dalam menyampaikan story dan informasi penting dari sebuah data.

Pada proses ini kita bisa meminta teman atau rekan setim untuk melihat seluruh visualisasi data yang telah kita buat. Selanjutnya, mintalah mereka untuk membuat kesimpulan dari seluruh visualisasi data yang telah dibuat. Lalu, pastikan kesimpulan mereka sesuai dengan yang ingin Anda sampaikan. Pada prosesnya, Anda bisa mengajukan beberapa pertanyaan berikut.

* Informasi apa yang Anda peroleh dari setiap visualisasi data tersebut?
* Menurut Anda apa kesimpulan utama dari seluruh visualisasi data tersebut?
* Apakah Anda memiliki pertanyaan terkait data tersebut yang belum terjawab?
* Apakah ada bagian yang belum Anda pahami dari seluruh visualisasi data tersebut?
* Berapa lama waktu yang Anda butuhkan untuk memahami setiap visualisasi data tersebut?

Anda dapat menggunakan beberapa pertanyaan tersebut guna mendapatkan feedback atau masukan untuk meningkatkan kualitas visualisasi data yang dibuat. Selain itu, pertanyaan tersebut juga membantu Anda mengidentifikasi informasi penting lain yang mungkin Anda lewatkan.

## Rangkuman Pengembangan Dashboard

### Pengenalan Dashboard

Saat ini, dashboard banyak digunakan sebagai media untuk memonitori berbagai metrik yang dianggap penting untuk perkembangan sebuah perusahaan. Selain itu, ia juga merupakan salah satu komponen penting dalam kultur data-driven decision making. Umumnya, sebuah dashboard terdiri dari dari beberapa visualisasi data yang interaktif dan saling berhubungan satu sama lain.

Dashboard yang efektif merupakan kunci utama dalam kultur data-driven decision making. Berikut merupakan beberapa hal yang harus diperhatikan dalam membuat dashboard yang efektif.

* **Memahami audiens**  
  Untuk membuat dashboard yang efektif, kita harus mempertimbangkan audiens dan kebutuhannya. Oleh karena itu, pada proses pembuatan dashboard kita harus memahami siapa target user atau audiens serta goals atau kebutuhan yang ingin mereka capai dari dashboard tersebut.
* **Pertimbangkan ukuran tampilan**  
  Sejalan dengan poin pertama, kita juga harus mempertimbangkan ukuran tampilan dashboard yang sesuai dengan device atau perangkat yang sering digunakan oleh target user atau audiens untuk melihat dashboard yang Anda buat.
* **Manfaatkan sweet spot**  
  Anda bisa menggunakan sweet spot untuk menampilkan informasi penting dari sebuah dashboard. Tentunya Anda perlu melakukan riset terlebih dahulu untuk mengetahui sweet spot yang tepat untuk kebutuhan Anda.
* **Pertimbangkan load time**  
  Load time merupakan salah hal yang penting untuk kita pertimbangkan guna menjaga pengalaman user atau audiens. Oleh karena itu, kita harus memastikan dashboard yang dibuat memiliki load time yang relatif singkat.
* **Batasi komponen visual yang ditampilkan**  
  Pada proses pengembangan dashboard kita juga harus mempertimbangkan jumlah komponen visual yang akan ditampilkan. Secara umum, kita disarankan untuk hanya menampilkan dua atau tiga visualisasi data serta menghindari penggunaan komponen visual yang tidak mendukung dalam penyampaian informasi penting.

### Pengenalan Streamlit

Streamlit merupakan salah satu open-source web app framework untuk bahasa pemrograman Python. Ia memungkinkan kita membuat web app yang baik dan interaktif dalam waktu yang singkat. Selain itu, ia juga kompatibel dengan berbagai library populer seperti NumPy, pandas, matplotlib, dll. Inilah yang menjadi alasan streamlit sering digunakan oleh oleh para praktisi data dan machine learning.

Seperti biasa, sebelum mulai menggunakan streamlit tentunya kita perlu menyiapkan environment-nya terlebih dahulu. Berikut merupakan tahapan untuk melakukannya.

1. Untuk menggunakan streamlit, kita perlu menulis kode Python dalam berkas**.py**. Oleh karena itu, Anda perlu menginstal code editor, seperti [Visual Studio Code](https://code.visualstudio.com/download), [Sublime Text](https://www.sublimetext.com/3), atau [PyCharm](https://www.jetbrains.com/pycharm/download/#section=windows) untuk mempermudah dalam menulis kode.
2. Selanjutnya, Anda perlu mengaktifkan virtual environment sebelumnya yang telah kita gunakan dalam materi latihan. Aktifkan virtual environment dengan menjalankan perintah berikut.
   1. conda activate main-ds
3. Tahap berikutnya adalah menginstal streamlit. Berikut merupakan perintah yang bisa Anda gunakan untuk menginstal streamlit.
   1. pip install streamlit

### Basic Element dalam Streamlit

Sebagai sebuah web app framework yang andal, streamlit telah menyediakan banyak pilihan element, widget, layout serta container untuk memastikan kita dapat membuat web app atau dashboard yang menarik dan interaktif.

#### Write

Basic element pertama yang akan kita bahas ialah write. Ia merupakan elemen streamlit yang digunakan untuk menampilkan sebuah output.  Untuk menggunakan element ini, kita hanya perlu memanggil function **write()** dan diikuti inputan yang ingin ditampilkan.

#### Text

Elemen lain yang ada dalam streamlit ialah text. Sesuai dengan namanya, ia merupakan elemen yang digunakan untuk menampilkan sebuah output berupa text. Elemen ini memiliki banyak function yang bisa digunakan sesuai kebutuhan. Berikut merupakan beberapa pilihan yang tersedia saat ini.

* **markdown()**  
  Function ini digunakan untuk menampilkan output dari argument markdown.
* **title()**  
  Ini merupakan function yang digunakan untuk menampilkan teks dalam format judul.
* **header()**  
  Function ini digunakan untuk menampilkan output teks sebagai format header.
* **subheader()**  
  Ini merupakan function yang digunakan untuk menampilkan output teks sebagai format subheader.
* **caption()**  
  Ia merupakan function yang digunakan untuk menampilkan output teks dalam ukuran kecil. Function ini biasanya digunakan untuk menampilkan caption, footnotes, dll.
* **code()**  
  Pada beberapa case, kita harus menampilkan potongan kode ke dalam streamlit app (web app yang dibuat menggunakan streamlit). Untuk menjawab hal ini, streamlit telah menyediakan sebuah function bernama **code()**.
* **text()**  
  Function selanjutnya ialah **text()**. Function ini digunakan untuk menampilkan sebuah normal teks.
* **latex()**  
  Function terakhir yang dapat digunakan untuk menampilkan elemen teks ialah latex(). Sesuai namanya, function tersebut digunakan untuk menampilkan mathematical expression yang ditulis dalam [format LaTeX](https://katex.org/docs/supported.html).

#### Data Display

Basic element selanjutnya yang akan kita bahas ialah data display. Ia merupakan elemen yang digunakan untuk menampilkan data secara cepat dan interaktif ke dalam streamlit app yang kita buat. Elemen ini memiliki beberapa function seperti berikut.

* **dataframe()**  
  Ia merupakan function yang digunakan untuk menampilkan DataFrame sebagai sebuah tabel interaktif. Pada function ini, kita bisa mengatur ukuran dari table yang ingin ditampilkan menggunakan parameter **width** dan **height**.
* **table()**  
  Selain **dataframe()**, kita juga bisa menggunakan function **table()** untuk menampilkan data ke dalam streamlit app. Ia dapat digunakan untuk menampilkan data dalam bentuk static table.
* **metric()**  
  Ketika membuat dashboard, terkadang kita perlu menampilkan sebuah metrik tertentu. Untuk melakukan hal ini, kita bisa memanfaatkan function **metric()**.
* **json()**  
  Selain bentuk DataFrame atau tabel, terkadang kita juga perlu menampilkan data dalam format JSON. Streamlit telah menyediakan function json() untuk menampilkan data dalam format JSON.

#### Chart

Basic element terakhir yang perlu kita ketahui ialah chart. Sesuai namanya, elemen ini dapat digunakan untuk menampilkan grafik visualisasi data ke dalam streamlit app. Elemen inilah yang akan sering kita gunakan untuk membuat dashboard nantinya.

Function **pyplot()** dapat digunakan untuk menampilkan grafik visualisasi data yang dibuat menggunakan matplotlib.

### Basic Widgets dalam Streamlit

Untuk menghasilkan sebuah dashboard yang interaktif, kita memerlukan komponen lain seperti widget (elemen Graphical User Interface yang memungkinkan pengguna untuk berinteraksi dengan aplikasi).

#### Input Widget

Kategori widget yang akan kita bahas ialah input widget. Ia merupakan kategori widget yang memungkinkan pengguna untuk memberikan input ke dalam streamlit app. Terdapat beberapa widget yang termasuk kategori ini, seperti text input, number input, date input, dll. Yuk, kita bahas satu per satu!

* **Text input**  
  Text input merupakan widget yang digunakan untuk memperoleh inputan berupa single-line text. Kita bisa menggunakan function **text\_input()** untuk membuat widget ini.
* **Text-area**  
  Text area merupakan widget yang memungkinkan pengguna untuk menginput multi-line text. Untuk membuat widget ini, streamlit telah menyediakan function bernama **text\_area()**.
* **Number input**  
  Ia merupakan widget yang digunakan untuk memperoleh inputan berupa angka dari pengguna. Anda dapat menggunakan function **number\_input()** untuk membuat number input widget.
* **Date input**  
  Selain inputan berupa angka dan teks, pada beberapa kasus kita juga membutuhkan input berupa tanggal dari pengguna melalui date input widget. Kita dapat membuat widget tersebut menggunakan function **date\_input()**.
* **File uploader**  
  Widget ini memungkinkan kita meminta pengguna untuk meng-upload sebuah berkas tertentu ke dalam web app. Kita dapat membuat file uploader widget menggunakan function  **file\_uploader()**.
* **Camera input**  
  Selain file uploader, streamlit juga menyediakan camera input widget yang dapat digunakan untuk meminta user mengambil gambar melalui webcam sekaligus mengunggahnya menggunakan function**camera\_input()**.

#### Button Widgets

Oke, kategori widget selanjutnya yang akan kita bahas ialah button widget. Ia merupakan kategori widget yang terdiri dari button, checkbox, radio button, dll.

* **Button**  
  Button merupakan widget untuk menampilkan tombol interaktif. Tombol tersebut dapat digunakan pengguna untuk melakukan aksi tertentu. Untuk membuat widget ini, kita bisa menggunakan function **button()**.
* **Checkbox**  
  Checkbox merupakan widget yang digunakan untuk menampilkan sebuah checklist untuk pengguna. Kita bisa menggunakan function **checkbox()** untuk membuat dan menampilkan checklist dalam streamlit app.
* **Radio button**  
  Ia merupakan widget yang memungkinkan pengguna untuk memilih satu dari beberapa pilihan yang ada. Untuk membuat widget ini, kita bisa menggunakan function **radio()**.
* **Select Box**  
  Select box merupakan widget yang memungkinkan pengguna untuk memilih salah satu dari beberapa pilihan yang ada. Ia merupakan opsi alternatif dari radio button. Streamlit telah menyediakan function **selectbox()** untuk membuat select box widget.
* **Multiselect**  
  Ia merupakan widget yang digunakan agar user dapat memilih lebih dari satu pilihan dari sekumpulan pilihan yang ada. Untuk mempermudah kita dalam membuat multiselect widget, streamlit telah menyediakan function bernama **multiselect()**.
* **Slider**  
  Slider merupakan widget yang memungkinkan pengguna untuk untuk memilih nilai (atau range nilai) dari sebuah range nilai yang telah ditentukan. Streamlit telah menyediakan function **slider()** untuk membuat slider widget.

### Basic Layouts dalam Streamlit

Untuk membuat dashboard yang rapi, kita perlu belajar cara mengatur layout atau tampilan dari sebuah dashboard. Sebagai sebuah web app framework, streamlit telah menyediakan berbagai pilihan layout yang dapat digunakan untuk mengatur tampilan web app (atau dashboard) yang akan dibuat.

#### Sidebar

Elemen layout pertama yang akan kita bahas ialah sidebar. Ia merupakan area yang berada di samping konten utama. Pada streamlit, posisi sidebar secara default berada di bagian kiri dari konten utama dan dapat memuat berbagai hal mulai dari gambar, teks, hingga widget.

Untuk menambahkan sebuah elemen atau widget ke dalam sidebar, kita bisa menggunakan notasi **with** yang diikuti sebuah object bernama **sidebar** yang telah disediakan oleh streamlit.

#### Columns

Columns merupakan elemen layout yang memungkinkan kita untuk mengatur tampilan pada konten utama ke dalam beberapa kolom sesuai kebutuhan.

Untuk membuat column dalam streamlit app, kita harus membuat object dari setiap kolom terlebih dahulu. Hal ini dapat dilakukan dengan memanfaatkan function **columns()**. Selanjutnya, kita hanya perlu menambahkan sebuah elemen atau widget ke dalam column tersebut menggunakan notasi **with**.

#### Tabs

Ia merupakan elemen layout yang memungkinkan kita untuk menambahkan beberapa tabs ke dalam konten utama. Hal ini tentunya akan sangat membantu kita dalam menghasilkan dashboard yang interaktif.

Sama halnya dengan columns yang sebelumnya kita bahas, untuk membuat tabs, kita harus membuat object dari setiap tab menggunakan function **tabs()** yang diikuti dengan list dari nama dari setiap tab.

#### Container

Container merupakan elemen layout dalam streamlit yang memungkinkan kita untuk membuat sebuah wadah untuk menampung suatu atau beberapa elemen dengan ukuran yang tetap. Container ini dapat kita gunakan untuk menghasilkan dashboard yang rapi.

Untuk membuat container, kita bisa menggunakan notasi **with** dan diikuti function **container()**.

#### Expander

Elemen layout selanjutnya yang tidak kalah penting ialah expander. Anda dapat menganggap elemen layout ini sebagai sebuah container yang dapat diperluas atau diciutkan secara interaktif.

Untuk membuat elemen layout ini, kita bisa menggunakan notasi **with** dan diikuti dengan function **expander()**.

## Rangkuman Kelas

## Rangkuman Dasar-Dasar Analisis Data

### Pengenalan Analisis Data

Data merupakan sekumpulan fakta yang dapat direpresentasikan dalam berbagai bentuk seperti angka, gambar, video, teks, hasil pengukuran, dll. Sebagai generasi kekinian, kita telah memproduksi banyak data setiap harinya. Selain menghasilkan banyak data, tanpa sadar kita juga sering memanfaatkan data untuk mengambil sebuah keputusan.

Analisis data dapat diartikan sebagai sebuah proses mengumpulkan, mentransformasi, dan menata data untuk menarik kesimpulan, membuat prediksi, serta memberi pertimbangan yang tepat dalam mengambil keputusan. Orang yang melakukan proses analisis data ini sering disebut sebagai data analyst.

Untuk mendukung proses analisis data, seorang data analyst membutuhkan beberapa keterampilan seperti berikut.

* **Analytical Skills**  
  Keterampilan ini berhubungan dengan kualitas dan karakteristik yang berhubungan dengan penyelesaian masalah menggunakan fakta. Berikut merupakan beberapa poin penting dalam analytical skill.
  + Memiliki rasa ingin tahu yang tinggi untuk belajar hal baru serta mencari pengalaman dan tantangan yang baru.
  + Memiliki kemampuan untuk memahami konteks atau kondisi di balik suatu kejadian.
  + Memiliki pola pikir teknis sehingga mampu menguraikan suatu hal menjadi langkah-langkah yg lebih kecil dan mengerjakan tiap bagiannya dengan cara yang teratur dan logis.
  + Mampu mengolah atau mengatur informasi secara baik dengan menerapkan data design.
  + Mampu menerapkan data strategy untuk manajemen orang, proses, dan tools yang digunakan dalam analisis data.
* **Analytical Thinking**  
  Hal ini melibatkan pengidentifikasian dan pendefinisian sebuah masalah, kemudian menyelesaikannya berdasarkan data dengan cara yang terorganisasi dan berurutan. Secara umum, terdapat enam aspek kunci dari analytical thinking antara lain sebagai berikut.
  + Merepresentasikan sebuah informasi dalam bentuk grafik atau visual. Hal ini akan membantu seorang data analyst untuk memahami dan menyampaikan sebuah informasi secara efektif.
  + Selalu menerapkan pola pikir strategis untuk menjaga fokus dengan membuat perencanaan matang.
  + Menerapkan pola pikir yang berorientasi pada masalah untuk mengidentifikasi, mendeskripsikan, dan menyelesaikan sebuah business problem.
  + Mengidentifikasi korelasi antara dua data atau lebih.
  + Menerapkan big-picture thinking (pemikiran gambaran besar) dalam proses analisis.
  + Mampu menerapkan detail-oriented thinking (pemikiran yang berorientasi pada detail) dalam proses analisis.

### Pentingnya Data dalam Dunia Bisnis

Di zaman yang serba canggih seperti saat ini, terdapat banyak inovasi dalam pemanfaatan teknologi dan informasi terutama pada bidang data. Data merupakan salah satu resource penting bagi perusahaan dari berbagai industri mulai dari e-commerce, entertainment, manufacturing, healthcare, marketing, finance, tech, dll.

Berbagai perusahaan tersebut membutuhkan data untuk mengidentifikasi peluang, tren, mengoptimalkan proses yang ada, meningkatkan layanan pelanggan, dan masih banyak lagi penerapannya di industri. Hal inilah yang mengakibatkan data dijuluki sebagai “The new oil” karena dianggap sebagai sumber daya paling penting saat ini.

Selain itu, insight (wawasan) yang diperoleh dari data dapat digunakan sebagai pendukung dalam pengambilan keputusan. Praktek seperti ini mendorong terbentuknya sebuah kultur baru dalam dunia bisnis yang disebut data-driven decision making.

Data-driven decision making merupakan sebuah kultur, best practice, serta proses dalam penggunaan data dan hasil analisis untuk memandu seluruh pengambilan keputusan bisnis yang tentunya sejalan dengan objektif perusahaan.

Umumnya perusahaan atau organisasi yang menerapkan kultur ini memiliki beberapa karakteristik seperti berikut.

* Memiliki data yang tersentralisasi dan terorganisir dengan baik.
* Memiliki kebijakan dan ketentuan terkait tata kelola data.
* Memiliki tingkat aksesibilitas data yang tinggi.
* Memiliki tool analisis yang mudah digunakan.

Berikut merupakan beberapa manfaat penerapan data-driven decision making.

* Manfaat utama dalam data-driven decision making ialah akuntabilitas dan konsistensi dalam setiap pengambilan keputusan. Hal ini akan mempermudah perusahaan dalam membuat sebuah benchmark yang jelas untuk mengevaluasi setiap keputusan dan strategi bisnis yang dibuat sehingga mampu mendorong perkembangan perusahaan secara kontinu.
* Pengambilan keputusan berdasarkan data akan mengarahkan perusahaan untuk berinovasi dan memperoleh peluang bisnis.
* Data-driven decision making mampu meningkatkan efisiensi dengan menyediakan berbagai informasi yang relevan dalam mengambil sebuah keputusan.

Terdapat beberapa poin yang harus diperhatikan ketika ingin menerapkan kultur data-driven decision making yaitu seperti berikut.

* Mengumpulkan data sebanyak mungkin
* Menentukan tujuan yang dapat diukur
* Memastikan Aksesibilitas Data
* Menentukan prioritas

### Mengenal Siklus Analisis Data

Seperti yang telah kita bahas bersama, analisis data merupakan sebuah proses mengumpulkan, mentransformasi, dan menata data untuk menarik kesimpulan, membuat prediksi, serta memberi pertimbangan yang tepat dalam mengambil keputusan. Sebagai suatu proses, analisis data tentunya memiliki beberapa tahapan seperti berikut.

1. **Mendefinisikan pertanyaan**  
   Pada awal proyek analisis data, kita perlu mendefinisikan berbagai pertanyaan analisis terkait data ataupun permasalahan yang ingin diselesaikan. Pertanyaan tersebut akan membantu kita dalam memahami masalah bisnis yang dihadapi serta tujuan yang ingin dicapai. Bukan hanya itu, dengan adanya pertanyaan tersebut, kita akan merasa terbantu karena bisa lebih fokus pada bagian data yang relevan dengan tujuan.  
     
   Pada proses analisis data, terkadang kita diberikan data terlebih dahulu lalu mulai mendefinisikan pertanyaan analisis berdasarkan data tersebut. Namun, tidak jarang kita juga berangkat dari pertanyaan analisis terlebih dahulu lalu mencari data yang sesuai untuk menjawab pertanyaan tersebut.
2. **Data wrangling**  
   Setelah kita memiliki pertanyaan analisis yang baik, tahap selanjutnya adalah data wrangling. Pada tahap ini, Anda memulai dengan mengumpulkan data yang dibutuhkan untuk menjawab pertanyaan analisis yang telah dibuat sebelumnya. Kemudian Anda perlu menilai kualitas dan struktur dari data tersebut. Hal ini dilakukan untuk mengidentifikasi masalah yang terdapat dalam data serta membuat strategi pembersihan data yang sesuai.  
     
   Proses pembersihan data umumnya dilakukan dengan memodifikasi, mengganti, atau menghilangkan data yang bermasalah sehingga diperoleh dataset yang berkualitas dan terstruktur dengan baik. FYI, pembersihan data merupakan salah satu proses yang paling menghabiskan waktu dalam proyek analisis data. Oleh karena itu, kita sangat disarankan untuk menyediakan waktu yang cukup untuk melakukan pembersihan data.
3. **Exploratory Data Analysis**  
   Tahap berikutnya adalah data exploratory data analysis. Pada tahap inilah Anda akan melakukan eksplorasi terhadap data yang telah dibersihkan untuk memperoleh insight dan menjawab pertanyaan analisis.  
     
   Proses eksplorasi umumnya dilakukan dengan memanfaatkan berbagai teknik descriptive statistics (Penggunaan konsep statistik untuk mendeskripsikan data). Hal ini bertujuan untuk menemukan pola, hubungan, serta membangun intuisi terkait data yang diolah.  
     
   Selain menggunakan descriptive statistic, terkadang kita juga menggunakan teknik machine learning atau inferential statistic untuk memprediksi data di masa depan. Namun, kedua teknik tersebut di luar cakupan pembahasan kita pada kelas ini.
4. **Data visualization**  
   Hasil yang diperoleh dari exploratory data analysis perlu divisualisasikan melalui tahap data visualization. Pada tahap ini, kita akan menerapkan berbagai teknik dan jenis visualisasi data yang tepat untuk mengomunikasikan temuan atau insight dari hasil analisis secara efektif.
5. **Draw conclusion & communicate**  
   Pada akhir proyek analisis data, kita perlu membuat conclusion atau kesimpulan dari hasil analisis. Kesimpulan yang dibuat haruslah menjawab semua pertanyaan dan tujuan yang telah didefinisikan di awal.  
     
   Selain membuat kesimpulan, kita juga perlu membagikan temuan atau insight dari hasil analisis. Terdapat banyak cara dalam mengomunikasikan hasil analisis, seperti menggunakan report, slide deck, dan atau presentation.

Di industri, tahapan tersebut tidak selamanya berbentuk liner melainkan ada beberapa tahapan yang berulang. Hal inilah yang membuat tahapan dalam analisis data lebih cocok disebut sebagai sebuah siklus karena terdapat banyak iteratif proses di dalamnya.

### Pengenalan Tool Pengolahan Data

Sebagai salah satu bahasa pemrograman kekinian, Python dapat digunakan untuk mengerjakan banyak tugas salah satunya adalah mengolah dan memvisualisasikan data.

Untuk mendukung proses pengolahan dan analisis data, Python memiliki banyak libraries dan tools yang bersifat open-source dan dapat digunakan secara bebas. Berikut merupakan tiga libraries yang paling sering digunakan untuk mengolah data.

* **Library NumPy**  
  NumPy (singkatan dari numerical Python) merupakan salah satu library Python yang sangat powerful untuk membuat dan mengolah multi-dimensional arrays (sering juga disebut sebagai matriks atau tensor). NumPy dibuat dengan tujuan untuk mempercepat proses komputasi numerik dalam bahasa pemrograman Python.
* **Library pandas**  
  Selain NumPy, library lain yang sering digunakan adalah pandas. Ia merupakan library Python yang spesifik digunakan untuk memanipulasi dan menganalisis data. Library inilah yang akan kita gunakan sebagai tool utama dalam menganalisis data.
* **Library SciPy**  
  Library penting selanjutnya adalah SciPy (singkatan dari Scientific Python). Ia merupakan library Python yang khusus digunakan untuk kebutuhan komputasi saintifik. Library ini memanfaatkan NumPy sebagai komponen dasar dalam setiap perhitungan komputasi yang akan dilakukan.

### Pengenalan Tool Visualisasi Data

Pada proses analisis data, selain mengolah data, kita juga perlu merepresentasikan data tersebut ke dalam bentuk visual yang menarik dan mudah dipahami. Nah, pada materi kali ini, kita akan berkenalan dengan berbagai tools visualisasi data yang ada dalam bahasa pemrograman Python.

Sebagai salah satu bahasa pemrograman paling populer di kalangan praktisi data, Python memiliki banyak libraries dan tools yang bersifat open-source untuk mendukung pembuatan visualisasi data yang baik dan menarik. Berikut merupakan dua library Python yang paling sering digunakan oleh praktisi data di industri.

* **Library matplotlib**  
  Matplotlib merupakan sebuah library Python yang reliable dan komprehensif untuk mendukung kebutuhan pembuatan visualisasi data. Matplotlib menyediakan banyak object dan method yang memungkinkan kita membuat visualisasi data statis, animated, dan interaktif.
* **Library seaborn**  
  Selain matplotlib, library lain yang sering digunakan untuk visualisasi data adalah seaborn. Seaborn merupakan library Python yang spesifik digunakan untuk membuat visualisasi data yang atraktif dan informatif. Library ini memanfaatkan matplotlib untuk menampilkan grafik visualisasi data.

## Rangkuman Penerapan Dasar-Dasar Descriptive Statistics

### Pengantar Penerapan Dasar-Dasar Statistik

By definition, statistik merupakan ilmu yang menerapkan berbagai metode saintifik dalam proses pengumpulan, analisis, interpretasi, serta penyajian data. Terdapat beberapa konsep statistik yang umum digunakan untuk mendeskripsikan sebuah data yang dikenal sebagai descriptive statistics. Beberapa konsep tersebut bisa Anda lihat di bawah ini.

* Measuring central tendency untuk mengidentifikasi nilai khas atau sentral dalam sebuah data.
* Measuring dispersion untuk menilai variabilitas atau sebaran nilai dalam sebuah data.
* Measuring asymmetric untuk menilai ketidaksimetrisan dalam sebuah distribusi data.
* Data relationship untuk mengidentifikasi hubungan dua atau lebih feature/column/variable dalam sebuah data.

#### Tipe Data

Pada proses analisis, tipe data akan menentukan metode analisis dan bentuk visualisasi data yang tepat. Secara garis besar, data dapat dibagi menjadi dua tipe, yaitu kuantitatif (numerik) dan kategoris.

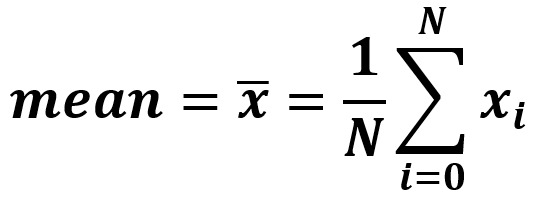
|  |  |  |  |
| --- | --- | --- | --- |
| Data kuantitatif | Data kontinu | Direpresentasikan dalam berbagai nilai numerik dan bisa dibagi atau diubah ke dalam unit yang lebih kecil | Umur, tinggi badan, berat badan, dll. |
| Data diskret | Direpresentasikan dalam bilangan bulat dan tidak dapat dibagi ke dalam unit yang lebih kecil | Jumlah kucing peliharaan, jumlah produk yang terjual, dll. |
| Data kategoris | Data ordinal | Tipe data kategoris yang dapat diurutkan berdasarkan rangking | Rating, kategori pelanggan berdasarkan income, dll. |
| Data nominal | Tipe data kategoris yang tidak memiliki urutan atau rangking | Jenis kelamin, warna bulu kucing, dll. |

### Measuring Central Tendency

Measuring central tendency merupakan sekumpulan parameter statistik untuk menggambarkan nilai khas atau sentral yang mewakili keseluruhan observasi atau data. Nilai khas atau sentral tersebut direpresentasikan menggunakan tiga parameter statistik yaitu mean, median, dan mode.

#### Mean

Mean atau sering juga disebut sebagai expected value merupakan salah satu parameter yang paling sering digunakan untuk menggambarkan nilai khas atau sentral dari suatu data. Ia diperoleh dari menghitung jumlah keseluruhan data dan dibagi dengan banyaknya data yang dimiliki.

[](https://www.dicoding.com/academies/555/tutorials/31220?from=31215)

Parameter ini sangat sensitif terhadap noise dan outlier yang terdapat dalam sebuah data.  Inilah salah satu alasan kenapa kita tidak boleh hanya menggunakan mean untuk merepresentasikan nilai khas dari suatu data.

#### Median

Median merupakan parameter yang merepresentasikan nilai tengah atau persentil ke-50 dari keseluruhan observasi atau data. Parameter ini diperoleh dengan mengambil nilai persentil ke-50 dari data yang telah diurutkan.

Untuk data yang berjumlah ganjil, nilai median diperoleh dengan mengambil langsung nilai tengah dari data yang telah diurutkan. Namun, hal ini berbeda jika data yang dimiliki berjumlah genap. Pada kasus seperti ini, kita perlu mengurutkan dan menghitung nilai rata-rata dari dua data yang berada di tengah

Median dikenal sebagai salah satu parameter yang sangat toleran terhadap outlier. Apabila Anda mengubah salah satu nilai dari data tersebut, parameter median yang diperoleh tidak akan berubah secara signifikan.

#### Mode

Parameter ketiga dalam measuring central tendency ialah mode. Parameter ini akan memberikan kita gambaran atau informasi terkait nilai yang paling sering muncul dalam suatu data.

Untuk menghitung parameter mode, kita perlu mengurutkan nilai dari suatu data terlebih dahulu. Selanjutnya memilih nilai yang paling sering muncul. Sebagai contoh, pada data jumlah kucing peliharaan yang dimiliki pegawai Dicoding, nilai yang paling sering muncul adalah 2. Nilai ini merupakan mode dari data tersebut.

Ketika menggunakan mode, Anda perlu memahami bahwa parameter ini akan mencari nilai yang paling sering muncul. Oleh karena itu, jika suatu data memiliki nilai yang unik, data tersebut tidak memiliki mode. Selain itu, apabila data memiliki lebih dari satu nilai yang paling sering muncul, itu artinya data tersebut memiliki lebih dari satu mode.

### Measuring Dispersion

Pada materi kali ini, kita akan fokus membahas sebuah konsep statistik untuk menilai sebaran dari suatu data yaitu measuring dispersion atau sering juga disebut measures of spread. Ia akan memberikan informasi terkait seberapa jauh suatu titik data dengan titik data lainnya. Hal ini mampu memberikan kita gambaran variasi nilai dari data tersebut. Untuk menggambarkan measuring dispersion, kita dapat menggunakan empat parameter, yaitu range, interquartile range, variance, dan standard deviation.

#### Range

Parameter pertama yang akan kita bahas adalah range. Ia merupakan parameter yang digunakan untuk melihat perbedaan antara nilai maksimum dan minimum dari suatu data.

Untuk menghitung parameter ini, kita perlu mengidentifikasi nilai maksimum dan minimum dari suatu data terlebih dahulu. Selanjutnya, kita mengurangi nilai maksimum dan nilai minimum. Selisih inilah yang menjadi parameter range.

#### Interquartile Range

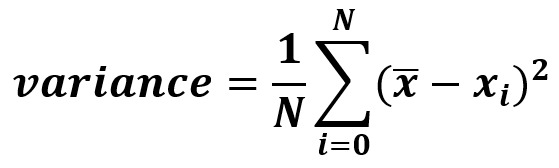
Interquartile Range atau sering disingkat IQR merupakan parameter statistik yang menggambarkan selisih antara kuartil ketiga (Q3) dan kuartil pertama (Q1). Parameter ini mewakili nilai range atau rentang di mana sebagian besar titik data berada.

Untuk menghitung IQR, kita perlu menentukan nilai Q1 dan Q3 terlebih dahulu. Selanjutnya, kita dapat menghitung selisih keduanya. Selisih inilah yang menjadi nilai dari parameter IQR.

#### Variance

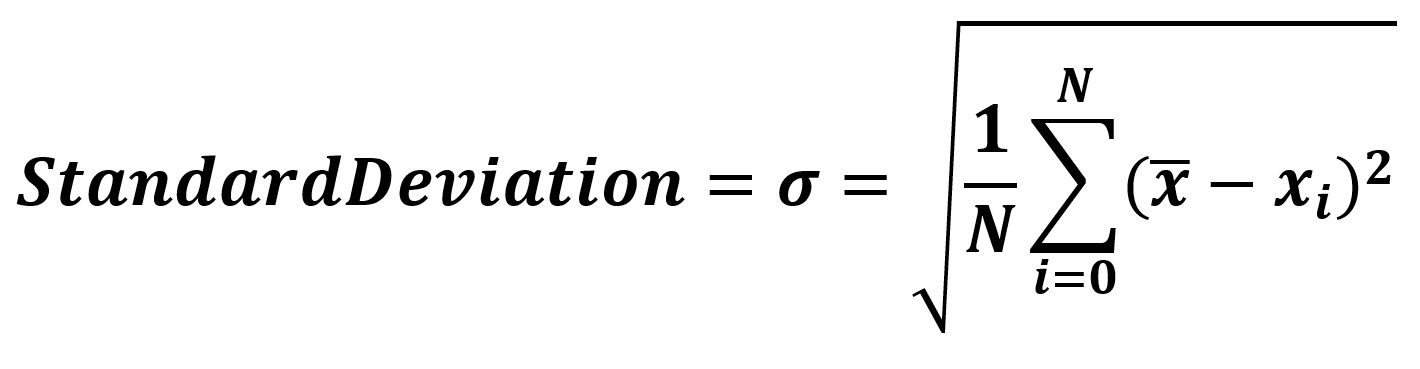
Selain range dan IQR, sebaran dari suatu data juga dapat direpresentasikan menggunakan parameter variance. Ia merupakan parameter yang digunakan untuk menggambarkan besar simpangan suatu titik data dari nilai mean-nya.

Sebelum menghitung nilai variance, kita membutuhkan nilai mean terlebih dahulu. Berikutnya kita perlu menghitung total selisih kuadrat dari seluruh titik data dengan nilai mean-nya. Nilai total tersebut selanjutnya akan dibagi dengan banyak titik data yang dimiliki. Berikut merupakan rumus untuk menghitung variance.

[](https://www.dicoding.com/academies/555/tutorials/31220?from=31215)

#### Standard Deviation

Jika Anda perhatikan, pada perhitungan variance kita sebenarnya telah menghitung rata-rata selisih kuadrat antar titik data dan nilai mean-nya. Penggunaan selisih kuadrat ini membuat unit dari parameter variance berbeda dengan unit data sehingga cukup sulit untuk diinterpretasikan. Oleh karena itu, kita membutuhkan satu parameter lain yang akan menghitung akar kuadrat dari variance. Parameter tersebut bernama standard deviation.

[](https://www.dicoding.com/academies/555/tutorials/31220?from=31215)

Standard deviation merupakan parameter yang paling sering digunakan para praktisi data untuk menilai sebaran atau simpangan dalam sebuah data. Semakin kecil nilai standard deviation, semakin kecil pula jarak antar titik data dengan nilai mean-nya sehingga bisa disimpulkan data tersebut memiliki sebaran yang sempit. Di sisi lain, semakin besar nilai standard deviation, semakin luas pula sebaran datanya. Selain itu, standard deviation juga sering digunakan untuk menilai seberapa baik parameter mean dalam merepresentasikan suatu data.

### Measuring Asymmetric

Measuring asymmetric merupakan konsep statistik yang digunakan untuk mengidentifikasi ketidaksimetrisan dalam sebuah distribusi data numerik. Parameter statistik yang umum digunakan dalam measuring asymmetric ialah skewness.

#### Distribusi Data

Distribusi data atau data distribution merupakan sebuah konsep statistik yang digunakan untuk menunjukkan frekuensi suatu nilai muncul dalam sebuah data. Umumnya kita menggunakan grafik histogram untuk melihat distribusi suatu data. Grafik tersebut akan menunjukkan frekuensi kemunculan setiap nilai dalam sebuah data.

Secara umum, berdasarkan bentuk grafik histogram yang terbentuk, kita bisa membagi distribusi data dalam tiga bentuk yaitu symmetric distribution, right-skewed distribution, dan left-skewed distribution.

* **Symmetric distribution**  
  Symmetric distribution (normal distribution atau bell curve distribution) merupakan suatu distribusi data yang memungkinkan kita untuk membagi data menjadi dua bagian secara simetris. Distribusi data ini memiliki nilai mean, median, serta mode yang sama dan berada di sentral distribusi data.
* **Right-skewed distribution**  
  Right-skewed distribution merupakan distribusi data yang memiliki sebagian besar populasi data yang terkonsentrasi pada bagian kiri. Distribusi data ini memiliki nilai mean lebih besar dari nilai median dan juga mode.
* **Left-skewed distribution**  
  Left-skewed distribution merupakan distribusi data yang terjadi ketika sebagian besar populasi data berada pada bagian kanan. Umumnya distribusi ini memiliki nilai median dan mode yang lebih besar dari nilai mean.

#### Skewness

Skewness merupakan parameter statistik yang digunakan untuk mengukur kesimetrisan sebuah distribusi data. Ia mampu memberikan kita gambaran tentang banyaknya data yang menyimpang dari symmetric distribution. Parameter ini memiliki jangkauan nilai mulai dari negatif hingga positif.

* Nilai skewness positif menggambarkan data yang memiliki distribusi yang cenderung right-skewed.
* Nilai skewness nol menggambarkan data yang memiliki distribusi simetris sempurna.
* Nilai skewness negatif merepresentasikan data dengan distribusi yang cenderung left-skewed.

### Data Relationship

By definition, data relationship merupakan konsep statistik untuk mengidentifikasi hubungan antar feature bertipe numerik dalam sebuah data. Untuk mengidentifikasi hubungan antar feature bertipe numerik, kita bisa menggunakan dua parameter yaitu correlation dan covariance.

#### Correlation

Parameter pertama yang akan kita bahas untuk mengidentifikasi data relationship ialah correlation. Parameter ini digunakan untuk mengidentifikasi korelasi atau hubungan dari dua feature numerik dalam sebuah data. Korelasi ini digambarkan menggunakan nilai dengan rentang -1 hingga 1.

Pada parameter correlation, nilai negatif menggambarkan korelasi berlawanan, sedangkan nilai positif merepresentasikan korelasi bersesuaian. Jika correlation dari dua feature bernilai nol, keduanya dinyatakan tidak memiliki korelasi.

#### Covariance

Selain menggunakan correlation, kita juga bisa menggunakan parameter covariance untuk mengidentifikasi hubungan antar dua feature dalam sebuah dataset. Berdasarkan nilainya, covariance terbagi menjadi tiga bagian seperti berikut.

* Positive covariance yang menggambarkan dua feature yang berkorelasi positif atau bersesuaian.
* Negative covariance yang merepresentasikan dua feature yang berkorelasi negatif atau berlawanan.
* Zero covariance yang menandakan dua feature yang tidak berkorelasi satu sama lain.

## Rangkuman Petimbangan dalam Pengolahan Data

### Pengantar Petimbangan dalam Pengolahan Data

Mungkin sebagian dari Anda pernah mendengar tentang beberapa dataset akademik yang umum digunakan sebagai bahan latihan para calon praktisi data masa depan, seperti iris dataset, titanic dataset, dll. Seluruh dataset tersebut memang dikhususkan sebagai bahan latihan dan benchmark dalam penelitian sehingga bentuknya sangat sederhana. Selain itu, dataset seperti ini umumnya sudah dalam keadaan bersih dan siap untuk digunakan dalam proses analisis data ataupun proses lanjutan lainnya.

Data yang kita jumpai di industri sangatlah kompleks dan umumnya masih sangat kotor sehingga perlu dilakukan tahap pembersihan data (data cleaning) terlebih dahulu sebelum digunakan. Selain itu, pada proses analisis data, kita sering kali harus menggunakan lebih dari satu dataset dengan ukuran data yang sangat besar untuk menjawab sebuah masalah bisnis.

Tantangan lain yang akan Anda jumpai di industri salah satunya adalah kompleksitas dari kebutuhan bisnis. Kebutuhan bisnis di industri umumnya dipengaruhi banyak hal, mulai dari keadaan pasar, keberadaan kompetitor, hingga keadaan makro ekonomi dan tentunya masih banyak lagi faktor yang lain. Semua faktor tersebut membuat kebutuhan bisnis di industri menjadi sangat dinamis.

Selain tantangan terkait kebutuhan bisnis dan data, seorang praktisi data juga dihadapkan dengan banyak tantangan lain seperti privasi dan etika dalam pengolahan data serta keamanan dari data itu sendiri. Saat ini, ada banyak regulasi yang mengatur kedua hal tersebut dan sebagai calon praktisi data yang Andal, kita harus mematuhi semua regulasi tersebut.

### Bias dalam Data

By definition, bias merupakan sebuah prasangka dan kecenderungan yang mendukung atau menentang suatu hal, individu, atau kelompok lain dengan cara yang kurang adil. Ia bisa terjadi secara sengaja maupun tidak sengaja. Bias merupakan salah satu masalah yang sering dijumpai di kehidupan sehari-hari termasuk dalam sebuah data.

Bias dalam data umumnya didefinisikan sebagai eror atau galat yang secara sistematis mengarahkan hasil menuju arah tertentu. Bias dalam data umumnya muncul disebabkan banyak faktor salah satunya ialah kesalahan dalam proses sampling.

Secara umum bias dalam data dapat dikategorikan dalam empat tipe, yaitu sampling bias, observer bias, interpretation bias, dan confirmation bias.

* **Sampling bias**  
  Sampling bias terjadi ketika sampel tidak mewakili populasi secara keseluruhan. Ini merupakan salah satu tipe bias yang paling sering terjadi. Bias seperti ini dapat dihindari menggunakan teknik random sampling ketika mengumpulkan sebuah data. Teknik ini memungkinkan semua bagian dari populasi mendapat kesempatan yang sama untuk terpilih.
* **Observer bias**  
  Tipe bias selanjutnya ialah observer bias atau sering juga disebut sebagai experimenter bias. Bias tipe ini muncul karena adanya kecenderungan yang berbeda dari setiap individu ketika melakukan observasi. Hal ini bisa dihindari dengan melakukan observasi berulang. Sebagai contoh, untuk menghindari observasi bias ketika melakukan pengukuran, kita bisa melakukan proses pengukuran secara berulang dan mengambil nilai rata-ratanya sebagai hasil akhir.
* **Interpretation bias**  
  Interpretation bias merupakan bias yang terjadi karena adanya kecenderungan kita dalam menginterpretasikan situasi ambigu hanya ke dalam dua keadaan (hitam dan putih). Hal ini berpotensi melahirkan bias karena setiap individu memiliki kecenderungan yang berbeda dalam menginterpretasi situasi yang ambigu. Hal ini dapat dihindari dengan cara melibatkan orang lain untuk mengonfirmasi hasil interpretasi yang kita buat.
* **Confirmation bias**  
  Tipe bias terakhir adalah confirmation bias. Bias ini muncul karena adanya kecenderungan kita dalam mencari atau menafsirkan informasi untuk mengonfirmasi keyakinan yang sudah ada sebelumnya. Tipe bias ini akan sangat berbahaya karena dapat menjebak kita dalam echo chamber. Oleh karena itu, sebagai calon praktisi data yang andal, kita harus selalu menempatkan diri senetral mungkin dalam setiap proses analisis data.

### Data Credibility

Kredibilitas data merupakan bagian dari data quality assessment yang digunakan sebagai sebuah tolok ukur untuk menilai tingkat kepercayaan terhadap suatu data.

Tingkat kepercayaan dari sebuah data akan sangat berdampak terhadap kualitas dan keakuratan hasil analisis yang akan dibuat. Tingkat kepercayaan ini tentunya dipengaruhi oleh banyak faktor, seperti sumber data, kelengkapan data, keakuratan data, dll.

Berikut merupakan beberapa poin sebagai patokan untuk memastikan data yang digunakan berkualitas dan dapat dipercaya.

* **Reliabilitas data**  
  Reliabilitas merupakan salah satu faktor penting untuk menilai kualitas dan kredibilitas dari sebuah data. Data disebut reliabel jika memuat informasi yang akurat, lengkap, dan tidak bias.
* **Orisinalitas data**  
  Selain reliabilitas, orisinalitas dari sebuah data juga merupakan komponen kunci dalam menilai kualitas dan kredibilitas dari sebuah data. Hal ini erat kaitannya dengan ownership (kepemilikan) dan sumber data.
* **Komprehensif**  
  Poin selanjutnya yang tidak kalah penting ialah komprehensif. Poin ini sangat penting untuk memastikan data yang digunakan memiliki semua informasi yang dibutuhkan untuk menjawab pertanyaan dan tantangan bisnis.
* **Aktualitas data**  
  Last, but not least adalah aktualitas data. Seperti yang telah kita bahas sebelumnya, data di industri umumnya bersifat dinamis atau terus berubah. Jika kita menggunakan data lama, informasi di dalamnya mungkin akan outdated sehingga berdampak terhadap hasil analisis yang dibuat. Oleh karena itu, sangat penting bagi kita untuk memastikan proses analisis dilakukan menggunakan data terbaru.

### Privasi dan Etika dalam Pengolahan Data

Pada beberapa dekade terakhir, kedua topik tersebut telah menjadi salah satu pembahasan hangat dalam dunia data. Hal ini merupakan imbas dari maraknya pemanfaatan data di industri sehingga mulai banyak yang menyadari potensi bahaya dalam praktik tersebut seperti pelanggaran privasi.

Untuk mencegah bahaya tersebut, mulai muncul banyak studi terkait data ethics. Ia merupakan kumpulan acuan standar dalam menilai benar dan salah sebuah proses pengolahan data. Berikut merupakan beberapa aspek penting dalam data ethics.

* **Ownership**  
  Aspek ini menyatakan bahwa kepemilikan data berapa pada setiap individu yang menghasilkan data tersebut. Sebagai pemilik data, individu memiliki kontrol penuh akan seluruh proses dan penggunaan data tersebut.
* **Transparency**  
  Seluruh tahapan dan algoritma dalam proses pengolahan data harus transparan dan dipahami oleh setiap individu yang menghasilkan data tersebut.
* **Consent**  
  Setiap individu harus mengetahui secara eksplisit terkait penggunaan dan alasan suatu data dikumpulkan. Hal ini dilakukan sebelum individu tersebut menyetujui pengumpulan data tersebut.
* **Currency**  
  Jika data digunakan untuk transaksi keuangan, setiap individu yang menghasilkan data harus mengetahui dengan jelas detail terkait transaksi tersebut.
* **Privacy & Openness**  
  Organisasi yang mengumpulkan data bertanggung jawab dalam menjaga setiap informasi yang bersifat pribadi pada setiap proses pengolahan data. Selain itu, organisasi tersebut harus menyediakan akses terhadap penggunaan dan pemanfaatan data (non private) secara bebas.

Berikut merupakan beberapa regulasi yang dibuat untuk menjaga data privasi milik pengguna dan memastikan penerapan data ethics.

* General Data Protection Regulation (GDPR)
* Peraturan Menteri (Permen) No 20 Tahun 2016 tentang Perlindungan Data Pribadi (PDP)

### Keamanan Data

Data security dapat didefinisikan sebagai upaya dalam menjaga dan melindungi informasi digital (data) dari berbagai pihak yang tidak bertanggung jawab dalam seluruh proses pengolahannya. Hal ini mencakup keamanan fisik hardware (storage device) serta keamanan dari software atau aplikasi yang bersangkutan. Selain itu, ia juga mencakup seluruh kebijakan dan prosedur dalam sebuah organisasi.

Berikut merupakan dua teknik atau metode A.

* **Encryption**  
  Encryption merupakan teknik yang memanfaatkan algoritma tertentu untuk mengubah data menjadi bentuk unusable bagi individu atau aplikasi yang tidak memiliki akses terhadap algoritma tersebut. Pada penerapnnya, kita bisa menganggap algoritma tersebut sebagai key (kunci) untuk membalikkan (reverse) proses encryption.
* **Tokenization**  
  Metode ini memungkinkan kita untuk mengubah elemen tertentu dalam sebuah data menjadi sebuah data random yang berperan sebagai token. Pada metode ini, data asli beserta token mapping akan disimpan dalam lokasi yang berbeda. Hal ini memungkinkan kita membatasi akses terhadap data asli dan memastikan keamanannya dengan menyimpan data tersebut di dalam lokasi yang berbeda.

## Rangkuman Data Wrangling

### Pengenalan Data Wrangling

Data wrangling merupakan sebuah proses atau kumpulan kegiatan yang meliputi pengumpulan data (Gathering data), penilaian data (Assessing data), serta pembersihan data (Cleaning data) sebelum data digunakan dalam proses analisis data.

* **Gathering data**  
  Tahap data wrangling, dimulai dengan proses pengumpulan data. Pada proses ini kita akan mengumpulkan semua data yang dibutuhkan untuk menjawab semua pertanyaan atau masalah bisnis yang ingin kita hadapi.
* **Assessing data**  
  Setelah semua data yang dibutuhkan terkumpul, proses selanjutnya ialah penilaian terhadap data tersebut. Proses ini dilakukan untuk menilai kualitas dan struktur dari sebuah data. Selain itu, proses ini juga bertujuan untuk mengidentifikasi berbagai masalah yang terdapat dalam data, seperti missing value, unstandard value, dll.
* **Cleaning data**  
  Apabila pada proses sebelumnya kita menemukan masalah (missing value, outlier, dll.) yang terdapat di dalam sebuah data, masalah tersebut harus dibersihkan sebelum masuk tahap analisis data. Terdapat beberapa teknik yang dapat kita gunakan untuk membersihkan data. Seluruh teknik tersebut akan kita pelajari pada beberapa materi ke depan.

### Gathering Data

Proses pengumpulan data merupakan salah satu tahapan paling menantang dalam proyek analisis data. Bergantung pada proyek analisis data yang dikerjakan, data yang Anda butuhkan mungkin terdapat dalam berbagai sumber dan memiliki format yang berbeda-beda.

Salah satu skill yang harus dimiliki oleh seorang praktisi data ialah mengetahui sumber data yang tepat untuk mengumpulkan data yang dibutuhkan. Berikut merupakan beberapa sumber data yang bersifat publik dan sering digunakan oleh para praktisi data.

* **Kaggle**  
  Kaggle merupakan platform andalan bagi praktisi data untuk mencari dataset. Selain menyediakan dataset, kaggle juga menyediakan berbagai kompetisi dan challenge terkait data science dan machine learning. Kaggle juga memungkinkan para praktisi data dan machine learning saling berbagai kode dan pengetahuan terkait data serta machine learning.
* **UCI Machine Learning Repository**  
  UCI Machine Learning Repository merupakan sebuah repositori yang menampung berbagai dataset yang bersifat publik.
* **Google Dataset Search**  
  Google Dataset Search merupakan sebuah search engine yang disediakan oleh Google. Ia dibuat untuk mempermudah para praktisi data dan juga researcher dalam mencari dataset yang bersifat publik.
* **Satu Data Indonesia**  
  Tahukah Anda bahwa pemerintah Indonesia sebetulnya telah menyediakan sebuah platform bernama Satu Data Indonesia atau sering disingkat SDI.

Selain menggunakan berbagai sumber dataset publik tersebut, para praktisi data juga seringkali menggunakan data internal yang bersifat privat. Data seperti ini umumnya disimpan dalam sebuah sistem pengolahan database.

Pandas menyediakan beberapa function yang dapat digunakan untuk membaca atau mengakses data menjadi sebuah DataFrame. Saat ini, Pandas mendukung pembacaan atau pengaksesan data dari berbagai format berkas data dan sumber data seperti pada contoh di bawah ini.

* **Format berkas CSV**  
  Berkas CSV (Comma Separated Values) merupakan format berkas data tabel yang paling sering digunakan dan telah menjadi standar dalam industri. Pandas menyediakan sebuah function **read\_csv()**untuk membaca berkas CSV.
* **Format berkas XLSX atau XLS**  
  Berkas XLSX atau XLS merupakan format berkas spreadsheet yang dibuat menggunakan aplikasi Microsoft Excel. Pandas juga menyediakan function **read\_excel()** untuk membaca berkas data dengan format XLSX atau XLS.
* **Format berkas JSON**  
  Berkas JSON (JavaScript Object Notation) merupakan format berkas data yang memiliki struktur data yang mirip seperti data structure dictionary dalam Python yang terdiri dari pasangan keys dan values. Pandas juga menyediakan function **read\_json()** untuk membaca berkas data berformat JSON.
* **Format berkas HTML**  
  HTML atau dikenal juga sebagai HyperText Markup Language merupakan sebuah markup language standar yang digunakan untuk merancang tampilan sebuah dokumen/halaman di web browser. Untuk membaca berkas ini, pandas menyediakan function **read\_html()**.
* **Format berkas XML**  
  Format data selanjutnya yang akan kita bahas ialah XML yang merupakan singkatan dari Extensible Markup Language. Ia sering digunakan untuk merepresentasikan berbagai struktur informasi, seperti dokumen, data, konfigurasi, dll. Pandas menyediakan function **read\_xml()** untuk membaca format data ini.
* **Akses data dari SQL database**  
  Selain membaca data dari berbagai format, library pandas juga memungkinkan kita untuk mengakses data langsung dari sebuah database, seperti PostgreSQL, MySQL, dll. Tentunya untuk mengakses database tersebut kita membutuhkan library pendukung yaitu SQLAlchemy.  
    
  Untuk berinteraksi dengan database, pandas menyediakan tiga function seperti berikut.
  + **read\_sql\_table()** untuk membaca SQL database table dan mempresentasikannya ke dalam bentuk pandas DataFrame.
  + **read\_sql\_query()** untuk membaca SQL query dan mempresentasikannya ke dalam bentuk pandas DataFrame.
  + **read\_sql()** untuk membaca SQL query atau table dan mempresentasikannya ke dalam bentuk pandas DataFrame.

Seperti yang telah kita bahas sebelumnya, untuk menyelesaikan sebuah permasalahan bisnis, terkadang kita perlu menggunakan lebih dari satu tabel data. Tentunya untuk melakukan analisis kita perlu menggabungkan data tersebut. Salah satu teknik penggabungan data yang paling sering digunakan ialah merge atau join.

Berdasarkan cara penggabungannya, proses merge atau join dapat dibagi menjadi empat jenis yaitu seperti berikut.

* **Inner**  
  Inner join merupakan proses join yang hanya mengambil nilai yang bersesuaian di kedua tabel.
* **Left**  
  Left join merupakan proses join yang akan mengambil semua nilai dari tabel kiri beserta nilai yang bersesuaian dari tabel kanan.
* **Right**  
  Right join merupakan proses join yang akan mengambil semua nilai dari tabel kanan beserta nilai yang bersesuaian dari tabel kiri. Ia merupakan kebalikan dari left join.
* **Outer**  
  Outer join atau sering juga disebut full outer join merupakan proses join yang akan mengambil semua nilai dari kedua tabel. Ia merupakan gabungan dari left dan right join.

Sebagai tool andalan dalam pengolahan dan analisis data, pandas menyediakan sebuah function bernama **merge()**. Ia dapat digunakan untuk menggabungkan dua buah DataFrame.

### Assessing Data

Pemeriksaan data ini dilakukan dengan menjalankan proses assessing data. Ia merupakan proses yang bertujuan untuk mengidentifikasi masalah yang terdapat dalam data dan memastikan data tersebut berkualitas.

Seperti yang telah kita bahas sebelumnya, kebersihan dan kualitas dari sebuah data merupakan tantangan utama dalam proyek analisis data di industri. Data yang kotor umumnya memiliki masalah dalam kontennya. Berikut beberapa masalah yang umum dijumpai dalam sebuah data.

* **Missing value**  
  Masalah ini muncul karena adanya nilai yang hilang dari sebuah data dan biasanya direpresentasikan sebagai nilai **NaN** dalam library pandas. Hal ini biasanya terjadi karena adanya human error, masalah privasi, proses merging/join, dll.  
    
  Library pandas menyediakan sebuah method bernama **isnull()** atau**isna()** untuk mengidentifikasi missing value dalam sebuah DataFrame. Keduanya sering dipadukan dengan method **sum()** untuk menghitung total missing value pada setiap kolom dalam sebuah DataFrame.
* **Invalid value**  
  Masalah ini muncul ketika terdapat beberapa nilai yang tidak masuk akal, tidak sesuai dengan ketentuan, dan background knowledge dari data tersebut.  
    
  Untuk mendeteksi masalah seperti ini, kita membutuhkan teknik sedikit advance. Salah satu teknik yang paling sering digunakan ialah teknik filtering data menggunakan regex.
* **Duplicate data**  
  Duplicate data merupakan masalah lain yang umum dijumpai di industri. Ia terjadi ketika terdapat sebuah observasi (semua nilai dalam satu unit baris) yang memiliki nilai yang sama persis pada setiap kolomnya. Pandas menyediakan sebuah method **duplicated()** untuk mengidentifikasi apakah terdapat duplikasi pada sebuah DataFrame.
* **Inaccurate value**  
  Inaccurate value merupakan masalah yang muncul ketika nilai dalam sebuah data tidak sesuai dengan hasil observasi. Masalah ini umumnya muncul karena adanya human error atau sistem error.
* **Inconsistent value**  
  Inconsistent value adalah masalah yang muncul ketika sebuah data memiliki nilai yang tidak konsisten baik dari segi satuan maupun ketentuan penilaian. Inkonsistensi ini umumnya muncul karena adanya perbedaan standar dalam proses pengumpulan nilai.
* **Outlier**  
  Outlier atau dalam bahasa indonesia disebut pencilan merupakan titik data yang berada sangat jauh dari titik data yang lain dalam sebuah dataset. Nilai yang sangat jauh ini tentunya akan berdampak terhadap beberapa parameter statistik yang digunakan untuk menganalisis data, seperti nilai mean dan standard deviation.  
    
  Terdapat beberapa metode yang dapat digunakan untuk mengidentifikasi outlier dalam sebuah dataset. Metode yang paling sering digunakan ialah IQR method.  
    
  IQR method merupakan metode penentuan outlier berdasarkan nilai interquartile range (IQR). Ia mengidentifikasi outlier dengan cara membuat nilai cut-off sebagai faktor k (Umumnya kita menggunakan nilai 1.5 s/d 3) dari nilai IQR. Cut-off tersebut selanjutnya akan digunakan untuk menghitung nilai ambang batas (boundary values).  
    
  Metode lain yang bisa digunakan ialah box plot. Ia merupakan bentuk visual untuk merepresentasikan nilai IQR beserta ambang batas bawah dan atas dari sebuah data. Hal ini tentunya akan membantu kita mengidentifikasi outlier secara lebih mudah yaitu melalui bentuk visual.

### Cleaning Data

Data yang kotor juga harus kita bersihkan untuk memastikan ia tidak mempengaruhi hasil analisis yang akan kita lakukan nantinya.

Secara umum, proses pembersihan data dapat dibagi ke dalam tiga tahapan, yaitu define, code, dan test.

* **Define:** pada tahap ini, kita akan membuat rancangan tahapan serta metode pembersihan data berdasarkan masalah yang ditemukan dalam proses assessing data.
* **Code:** setelah membuat rancangan pembersihan data, tahap selanjutnya ialah mengonversi hal tersebut menjadi sebuah kode program yang dapat dijalankan.
* **Test:** setelah menjalankan kode program untuk membersihkan data, kita perlu memeriksa kembali data yang telah dibersihkan tersebut.

#### Teknik untuk Mengatasi Missing Value

Secara garis besar, terdapat tiga metode dalam mengatasi missing value antara lain seperti berikut.

* **Dropping**  
  Metode pertama dan yang paling mudah dalam mengatasi missing value adalah dropping. Pada metode ini, kita akan menghapus seluruh baris atau kolom yang memiliki missing value. Untuk melakukannya, kita bisa menggunakan salah satu method **dropna()** yang disediakan oleh library pandas  
    
  Metode dropping ini memang terkesan mudah, tetapi perlu Anda ketahui bahwa ada banyak sekali pertimbangan yang perlu diperhatikan sebelum menggunakan metode ini. Salah satu pertimbangan penting ialah penerapan metode ini dapat menyebabkan kita kehilangan banyak informasi.
* **Imputation**  
  Metode ini bekerja dengan cara mengisi (fill) missing value dengan nilai tertentu. Hal ini tentunya akan mencegah hilangnya informasi akibat missing value.  
    
  Pada data kontinu, kita bisa menggunakan nilai mean, median, atau mode sebagai pengganti missing value. Jika bekerja menggunakan data kategoris, kita dapat mengisi missing value dengan kategori yang paling sering muncul. Namun, perlu diingat bahwa pemilihan nilai pengganti ini harus didukung oleh background knowledge dari data tersebut.  
    
  Sebagai library andalan kita, pandas telah menyediakan sebuah method bernama **fillna()** untuk mengganti missing value dengan nilai tertentu.  
    
  Metode ini masih memiliki banyak kekurangan salah satunya ialah dapat mempengaruhi variance atau sebaran dari sebuah data. Selain itu, metode ini juga masih belum cukup baik untuk diterapkan pada data time series.
* **Interpolation**  
  Metode penanganan missing value terakhir yang akan kita bahas ialah interpolation (interpolasi). Sederhananya, interpolasi merupakan salah satu pendekatan numerik yang digunakan untuk menghitung titik data baru berdasarkan range data yang sudah ada. Perhitungan ini menggunakan sebuah persamaan garis linear ataupun polynomial. Library pandas juga menyediakan method **interpolate()** yang bisa kita gunakan untuk menerapkan metode interpolasi dalam mengatasi missing value.

#### Teknik untuk Mengatasi Outlier

Pada dasarnya, terdapat dua metode yang umum digunakan untuk mengatasi outlier yaitu drop dan imputation.

* **Drop**  
  Metode pertama yang paling mudah ialah men-drop atau menghapus seluruh baris yang mengandung outlier. Metode ini mampu mencegah outlier mempengaruhi hasil analisis yang kita buat.
* **Imputation**  
  Metode lain yang bisa Anda gunakan untuk menangani outlier ialah imputation. Konsepnya mirip seperti sebelumnya yaitu mengganti outlier dengan nilai tertentu. Nilai yang bisa kita gunakan ialah mean, media, dan mode. Selain itu, tidak jarang juga kita mengganti outlier dengan boundary value.  
    
  Untuk menerapkan metode ini, kita bisa menggunakan method **mask()** yang disediakan oleh library pandas. Method tersebut menerima parameter **cond** sebagai kondisi untuk memfilter nilai outlier.

#### Teknik untuk Mengatasi Duplicate Data

Ketika menemukan duplikasi pada data, tentunya kita harus menghilangkan atau menghapus duplikasi tersebut. Library pandas telah menyediakan sebuah method **drop\_duplicates()** untuk menghilangkan duplikasi dalam sebuah DataFrame.

## Rangkuman Exploratory Data Analysis

### Pengenalan Exploratory Data Analysis

Exploratory Data Analysis (EDA) merupakan tahap eksplorasi data yang telah dibersihkan guna memperoleh insight dan menjawab pertanyaan analisis. Pada prosesnya, kita akan sering menggunakan berbagai teknik dan parameter dalam descriptive statistics yang bertujuan untuk menemukan pola, hubungan, serta membangun intuisi terkait data yang diolah. Selain itu, tidak jarang kita juga menggunakan visualisasi data untuk menemukan pola dan memvalidasi parameter descriptive statistics yang diperoleh.

Exploratory analysis dan explanatory analysis merupakan dua istilah sering dijumpai dalam proyek analisis data.

Sederhana, exploratory analysis merupakan proses analisis data yang bertujuan untuk mengeksplorasi dan mengenal sebuah data. Pada prosesnya sering kali mulai dengan mendefinisikan berbagai pertanyaan atau hanya menggali data guna menemukan beberapa insight yang menarik dari sebuah data.

Di lain sisi, explanatory analysis merupakan proses analisis data yang bertujuan untuk membagikan beberapa insight yang menarik dari sebuah data. Insight inilah yang akan digunakan untuk menjawab permasalahan bisnis yang dihadapi.

Jika disimpulkan, explanatory analysis merupakan proses penyampaian temuan menarik dari proses exploratory analysis. Proses penyampaian ini tentunya harus diikuti dengan visualisasi data yang baik dan efektif.

### Mendefinisikan Pertanyaan Untuk Explorasi Data

Sebagai calon praktisi data masa depan, tentunya Anda telah mengetahui bahwa objektif dari sebuah proyek analisis data ialah menjawab tantangan atau permasalahan bisnis. Untuk memastikan pemahaman terhadap sebuah permasalahan bisnis, kita perlu mengurai masalah tersebut ke dalam beberapa pertanyaan analisis.

Problem solving merupakan salah satu keterampilan penting yang harus dimiliki seorang praktisi data. Untuk menyelesaikan sebuah problem, kita harus memahami terlebih dahulu permasalahan yang ingin diselesaikan. Berikut merupakan beberapa kategori problem yang sering dijumpai oleh para praktisi data.

* **Membuat prediksi**  
  Membuat prediksi merupakan salah satu problem yang paling sering dijumpai oleh praktisi data di industri. Jenis problem ini, mengharuskan kita untuk membuat sebuah prediksi berdasarkan hasil analisis dari data sebelumnya.
* **Mengategorikan berbagai hal**  
  Problem lain yang umum dijumpai di industri ialah kebutuhan untuk mengategorikan berbagai hal. Pada problem ini, kita harus menetapkan sebuah informasi ke dalam berbagai grup atau kluster berdasarkan kesamaan yang ada.
* **Menemukan sesuatu yang tidak biasa**  
  Kategori problem selanjutnya ialah mengidentifikasi atau menentukan sesuatu yang tidak biasa. Problem ini, umumnya muncul ketika terdapat sebuah kejadian aneh yang perlu dianalisis lebih lanjut.
* **Mengidentifikasi tema**  
  Permasalahan lain yang mungkin akan Anda temui ialah mengidentifikasi tema. Problem ini sering dijumpai dalam bidang UI/UX. Seorang UI/UX designer harus mengidentifikasi tema yang paling tepat dalam merepresentasikan anonymous survey data dari para pengguna. Hal ini bertujuan untuk meningkatkan pengalam pengguna ketika menggunakan sebuah produk.
* **Menemukan hubungan**  
  Kategori berikutnya ialah menemukan hubungan atau connection. Problem ini mengharuskan kita untuk mengidentifikasi dan menganalisis hubungan dari berbagai data untuk mengatasi suatu permasalahan.
* **Menemukan pola**  
  Last but not least, kategori problem yang mungkin Anda temui ialah menemukan pola. Jenis problem ini, membutuhkan analisis mendalam terhadap sebuah data guna memahami pola yang terdapat di dalamnya.

Jika data merupakan building block dalam proyek analisis data, kita dapat menganggap pertanyaan analisis sebagai kompas yang selalu mengarahkan untuk menuju goals dan objektif yang ingin dicapai. Pada proses pembuatan pertanyaan analisis, kita perlu menghindari beberapa poin berikut.

* Hindari membuat pertanyaan yang mengarahkan kita untuk menjawab dengan jawaban tertentu.
* Hindari membuat pertanyaan yang hanya dapat dijawab dengan “ya” atau “tidak.”
* Hindari membuat pertanyaan yang tidak memiliki konteks.

Untuk menghasilkan pertanyaan analisis yang efektif, terdapat beberapa poin yang harus Anda perhatikan seperti berikut.

* **Specific**  
  Pertanyaan yang efektif haruslah bersifat spesifik, artinya sebuah pertanyaan analisis harus fokus pada sebuah topik tertentu. Pertanyaan yang spesifik dapat membantu kita dalam mengumpulkan informasi yang relevan dengan tujuan dari proyek analisis data.
* **Measurable**  
  Selain bersifat spesifik, pertanyaan yang efektif haruslah bisa diukur atau measurable. Hal ini tentunya akan mempermudah kita dalam memahami dan menarik insight dari sebuah data.
* **Action-oriented**  
  Pertanyaan yang bersifat action-oriented berarti pertanyaan tersebut mampu memberikan informasi yang dapat membantu kita dalam membuat sebuah action-plan.
* **Relevant**  
  Pertanyaan yang relevan berarti pertanyaan tersebut berbobot dan memiliki arti untuk menyelesaikan sebuah permasalahan.
* **Time-bound**  
  Terakhir, pertanyaan yang efektif haruslah bersifat time-bound yang berarti pertanyaan tersebut haruslah memiliki relevansi terhadap waktu saat ini.

### Mengeksplorasi Parameter Statistik dari Data

Method **describe()** memungkinkan kita untuk melihat rangkuman parameter statistik (mean, median, dll.) dari kolom numerik pada sebuah DataFrame. Jika ingin melihat lebih banyak parameter statistik serta menyertakan kolom non-numerik, Anda bisa menggunakan parameter **include=”all”**.

Selain method **describe()**, pandas juga menyediakan method lain seperti **hist()** untuk membuat plot histogram dari kolom numerik pada sebuah DataFrame. Method ini dapat digunakan untuk memeriksa distribusi data secara cepat.

Jika ingin memeriksa korelasi antar data numerik pada sebuah DataFrame, Anda dapat menggunakan method **corr()** atau **cov()**.

Itulah beberapa method yang sering digunakan untuk mengeksplorasi parameter statistik dari sebuah DataFrame. Sebenarnya masih banyak method lain yang bisa Anda gunakan untuk memeriksa parameter statistik lain. Silahkan kunjungi dokumentasi berikut untuk melihat lebih detail terkait hal tersebut: [dokumentasi pandas.DataFrame](https://pandas.pydata.org/docs/reference/frame.html).

### Mengelompokkan Data

Mungkin beberapa dari Anda cukup familier dengan konsep pivot table dalam aplikasi pengolahan data seperti Microsoft Excel atau Google Sheets. Ia merupakan tool yang dapat digunakan untuk mengelompokkan data berdasarkan parameter tertentu.

Pada proses analisis data, kita bisa menggunakan konsep pivot table ini untuk menghitung parameter statistik dari kelompok data tertentu. Selain itu, ia juga dapat digunakan untuk kebutuhan visualisasi data dan pembuatan report.

Untuk memahami proses pembuatan pivot table, Anda perlu mengenal dua istilah di bawah ini telebih dahulu.

* Key: kolom yang berisi data kategorik dan digunakan untuk mengelompokkan data.
* Value: kolom yang berisi data kuantitatif dan akan dihitung nilai parameter statistiknya

Proses pembuatan pivot table, diawali dengan melakukan split atau proses pengelompokkan pasangan key-values. Tahap berikutnya adalah proses apply atau perhitungan parameter dari setiap kelompok pasangan key-value. Tahap terakhir ialah menggabungkan (combine) seluruh key dengan nilai parameternya ke dalam satu tabel. Tara…, jadilah sebuah pivot table yang akan membantu kita dalam memahami sebuah data.

Library pandas telah menyediakan sebuah method bernama groupby() yang dapat membantu kita dalam membuat pivot table secara lebih efisien dan scalable. Method ini akan menerima inputan beberapa parameter salah satunya ialah “by”. Parameter tersebut digunakan untuk menentukan key dari pivot table tersebut. Selain itu, untuk menggunakan method ini, kita perlu memadukannya dengan berbagai method untuk menghitung parameter statistik, seperti mean(), mode(), dll.

Untuk membuat pivot table dengan lebih dari satu parameter, kita bisa menggunakan method agg(). Ia memungkinkan kita untuk menentukan beberapa parameter yang ingin digunakan dalam pivot table.

Nah, itulah segenap pembahasan terkait pengelompokkan data atau pembuatan pivot table menggunakan method groupby().  Jika Anda ingin membaca lebih detail terkait method tersebut, silakan kunjungi tautan berikut: [GroupBy documentation](https://pandas.pydata.org/pandas-docs/stable/reference/groupby.html).

## Rangkuman Data Visualization

### Pengenalan Data Visualization

Pada siklus tersebut, data visualization merupakan tahapan yang harus kita lakukan sebelum membuat kesimpulan dan mengomunikasikan (draw conclusion & communicate) hasil dari proses analisis yang telah dilakukan. Hal ini dilakukan untuk mempermudah kita dalam memahami data dan membuat sebuah kesimpulan yang andal. Selain itu, visualisasi data juga dapat membantu kita dalam menyampaikan cerita atau temuan dari hasil analisis data kepada orang lain atau stakeholder.

Sederhananya, visualisasi data merupakan cara kita dalam menyajikan data dalam bentuk visual. Hal ini dilakukan untuk mempermudah kita dan orang lain dalam memahami data tersebut. Selain itu, visualisasi data yang baik juga akan sangat membantu kita dalam menyampaikan story dan pesan dari sebuah data.

### Prinsip-Prinsip dalam Visualisasi Data

Ketika berbicara tentang visualisasi data, kita tidak hanya membahas cara menyajikan data ke dalam bentuk visual, melainkan juga harus memahami berbagai prinsip dasar dalam membuat visualisasi data yang baik dan benar.

Umumnya visualisasi data yang buruk memuat salah satu dari tiga masalah berikut.

* **Misleading**  
  Masalah pertama dalam visualisasi data yang buruk ialah misleading information yang mampu mengakibatkan kesalahan dalam pengambilan kesimpulan dari sebuah data.
* **Hides**  
  Selain mislearning information, masalah lain yang umum dijumpai dalam visualisasi data yang buruk ialah menyembunyikan informasi tertentu.
* **Distracts**  
  Masalah lain yang mungkin terjadi ialah distraksi. Visualisasi data yang buruk sering kali menyertakan komponen visual yang seharusnya tidak dibutuhkan dan malah mengganggu proses pengambilan kesimpulan dari sebuah visualisasi data.

Jika ditelisik secara mendasar, visualisasi data merupakan proses dalam mengubah data ke dalam bentuk visual menggunakan berbagai elemen visual. Berikut merupakan beberapa elemen visual yang umum digunakan untuk membuat visualisasi data.

* **Position**: elemen ini akan membantu kita merepresentasikan titik data menggunakan sumbu tertentu (seperti sumbu X, Y, dan Z) sebagai acuan.
* **Size**: ukuran (panjang atau lebar) merupakan elemen visual yang umumnya kita gunakan untuk membedakan serta membandingkan nilai dari kategori atau titik data tertentu.
* **Shape**: bentuk merupakan salah satu elemen visual yang dapat digunakan untuk membedakan kategori atau titik data tertentu.
* **Color**: selain bentuk, warna juga merupakan pilihan elemen visual lain yang dapat digunakan untuk membedakan kategori atau titik data tertentu. Ketika menggunakan elemen ini, kita harus ingat bahwa tidak semua orang memiliki kemampuan untuk membedakan warna dengan baik.
* **Texture**: penambahan tekstur atau pola tertentu bisa menjadi alternatif lain dalam membedakan kategori atau titik data tertentu.
* **Angle**: pada beberapa pilihan bentuk visualisasi data, sudut merupakan salah satu elemen visual yang digunakan untuk merepresentasikan nilai dari suatu data.

Untuk memastikan penggunaan elemen visual secara efektif, Edward Tufte telah mengenalkan sebuah konsep bernama **Data Ink Ratio**pada bukunya yang berjudul “The Visual Display of Quantitative Data”.

Data ink ratio merupakan sebuah konsep parameter yang digunakan sebagai penentu efektivitas penggunaan visual elemen dalam sebuah visualisasi data. Ia didefinisikan sebagai perbandingan antara tinta (bisa diartikan sebagai elemen visual) yang digunakan untuk mendeskripsikan data dan total tinta yang digunakan dalam satu visualisasi data.

Ketika membuat sebuah visualisasi data, kita tidak hanya memikirkan terkait faktor estetika (seni) saja. Melainkan, kita juga harus memastikan visualisasi data yang kita buat mampu menyampaikan berbagai informasi penting dari sebuah data tanpa menyembunyikan fakta, mengarahkan orang lain pada asumsi tertentu, serta menghindari misinterpretasi.

Pada bukunya yang berjudul The Visual Display of Quantitative Data, Edward Tufte juga mengenalkan beberapa prinsip guna menjaga integritas dalam membuat visualisasi data. Berikut beberapa prinsip tersebut.

* Representasi angka pada sebuah grafik harus sebanding dengan kuantitas sebenarnya. Hal ini untuk menghindari terjadinya misinterpretasi dari sebuah visualisasi data.
* Pelabelan yang jelas, terperinci, dan menyeluruh merupakan kunci utama dalam menghindari ambiguitas dalam sebuah visualisasi data.
* Visualisasi data yang baik harus menunjukkan variasi pada data bukan variasi desain.
* Untuk menghindari asumsi, kita harus menggunakan unit yang umum digunakan (sesuai standar atau kesepakatan) dalam merepresentasikan suatu data tertentu.
* Jumlah variabel (feature/kolom) yang digambarkan tidak boleh melebihi jumlah dimensi dalam data. Sebagai contoh, untuk menggambarkan hubungan variabel X dan Y, kita disarankan untuk menggunakan plot dua dimensi ketimbang menggunakan plot tiga dimensi.
* Sebuah grafik tidak boleh menampilkan data yang di luar konteks.

Nah, itu dia keenam prinsip yang harus kita ikuti untuk menjaga integritas dan menghindari misinterpretasi dalam sebuah visualisasi data. Sebenarnya pada buku yang sama, Edward Tufte juga mengenalkan sebuah parameter bernama **Lie Factor** untuk mengukur kesalahan representasi dalam sebuah visualisasi data. Parameter ini didefinisikan sebagai rasio perbandingan ukuran yang ditampilkan pada grafik dan ukuran yang sebenarnya ada dalam data. Rasio ini tentunya akan memberikan kita gambaran sejauh mana visualisasi mendistorsi nilai dari sebuah data.

### Univariate Visualization

Seperti yang telah kita bahas sebelumnya, visualisasi data merupakan teknik yang digunakan dalam merepresentasikan atau menyajikan data ke dalam bentuk visual yang lebih mudah dipahami. Nah, berdasarkan jumlah variabel (feature/kolom) yang ingin disajikan, visualisasi data dapat dibagi menjadi tiga kategori.

* **Univariate visualization**  
  Univariate visualization merupakan bentuk visualisasi data yang hanya merepresentasikan informasi yang terdapat pada satu variabel. Jenis visualisasi ini umumnya digunakan untuk memberikan kita gambaran terkait distribusi sebuah variabel dalam suatu dataset.
* **Bivariate visualization**  
  Bivariate visualization merupakan bentuk visualisasi data untuk menyajikan informasi yang terdapat dalam dua variabel. Visualisasi jenis ini dapat digunakan untuk menganalisis hubungan antar dua variabel dalam sebuah dataset.
* **Multivariate visualization**  
  Last but not least, multivariate visualization merupakan jenis visualisasi data untuk menggambarkan informasi yang terdapat dalam lebih dari dua variabel. Jenis visualisasi ini digunakan untuk merepresentasikan hubungan dan pola yang terdapat dalam multidimensional data.

#### Bar Chart

Bentuk visualisasi data pertama yang akan kita bahas ialah bar chart atau dikenal juga dengan sebutan diagram batang. Ia merupakan pilihan visualisasi data yang digunakan untuk menggambarkan distribusi dari suatu data kategoris (categorical data).

Pada bar chart, setiap kategori akan direpresentasikan sebagai sebuah batang. Tinggi dari sebuah batang menunjukkan frekuensi atau jumlah titik data dari kategori tersebut.

Library matplotlib telah menyediakan sebuah function bernama **pyplot.bar()** untuk membuat sebuah bar chart. Untuk menggunakan function tersebut, kita perlu mendefinisikan parameter **x** dan **height** dari bar chart yang ingin dibuat.

Selain menggunakan library matplotlib, kita juga bisa membuat bar chart menggunakan library seaborn. Untuk membuat bar chart menggunakan library seaborn, kita bisa menggunakan function **barplot()**. Function ini akan menyediakan beberapa parameter penting seperti berikut.

* **data**: menampung DataFrame yang akan digunakan.
* **x, y**: menampung nama kolom atau data yang divisualisasikan.
* **orient**: orientasi dari bar chart yang akan digunakan (“v” atau “h”).
* **color**: mendefinisikan warna yang akan digunakan.

#### Pie Chart

Pie chart merupakan bentuk visualisasi data yang dapat digunakan untuk menggambarkan perbandingan frekuensi tiap kategori pada suatu data kategoris. Pada pie chart, setiap kategori digambarkan oleh irisan dalam sebuah lingkaran dan frekuensi digambarkan sebagai luasan (area) dari irisan tersebut.

Terdapat variasi lain dari pie chart, yaitu donut plot. Sebenarnya, ia merupakan pie chart yang memiliki lubang pada bagian tengahnya sehingga membentuk donut. Selain itu, tidak ada perbedaan mendasar dalam penggunaan antara pie chart dan donut plot.

Untuk membuat pie chart, kita bisa memanfaatkan library andalan, yaitu matplotlib. Ia telah menyediakan sebuah function bernama **pie()** untuk menghasilkan grafik pie chart. Function tersebut memiliki beberapa parameter penting yaitu seperti berikut.

* **x**: menampung data yang akan divisualisasikan.
* **explode**: menampung array atau list yang mengatur posisi tiap irisan lingkaran.
* **labels**: sekumpulan string yang digunakan untuk memberi label pada tiap irisan lingkaran.
* **colors**: sekumpulan warna yang akan digunakan pada tiap irisan lingkaran.
* **autopct**: string yang digunakan untuk memberi numerik label pada tiap irisan lingkaran.

#### Histogram

Jika dilihat sekilas, bentuk grafik histogram mirip dengan bentuk bar chart perbedaannya hanya terdapat pada sumbu X. Pada grafik histogram, sumbu X digunakan untuk menampung range nilai dari data kuantitatif yang dikenal dengan istilah bins.

Pada grafik histogram, ukuran bins akan sangat berpengaruh terhadap cara kita dalam merepresentasikan data kuantitatif. Jika menggunakan ukuran bins yang terlalu besar, kita akan kehilangan cukup banyak informasi dari data kuantitatif yang ingin divisualisasikan. Sebaliknya, apabila ukuran bins yang digunakan terlalu kecil, hal ini akan menghasilkan banyak noise yang mendistraksi distribusi data yang ingin kita amati.

Untuk membuat grafik histogram, kita bisa menggunakan function **hist()**yang disediakan oleh library matplotlib. Function ini menerima beberapa parameter seperti berikut.

* **x**: menampung data yang akan divisualisasikan.
* **bins**: menampung jumlah bins (sebanding dengan ukurannya) yang akan digunakan untuk membuat grafik histogram.

Selain menggunakan library matplotlib, kita juga bisa membuat grafik histogram dengan library seaborn. Library ini menyediakan function **histplot()** untuk membuat grafik histogram.

#### Box Plot

Bentuk visualisasi data lain yang bisa digunakan untuk mengidentifikasi distribusi pada data kuantitatif ialah box plot. Ia merupakan bentuk visual untuk merepresentasikan nilai IQR beserta ambang batas bawah dan atas dari sebuah data. Box plot ini juga bisa digunakan untuk mengidentifikasi outlier atau pencilan yang ada pada data kuantitatif.

Kita bisa menggunakan function**boxplot()** yang disediakan oleh library seaborn. Berikut contoh kode untuk menggunakannya.

### Bivariate dan Multivariate Visualization

Sederhananya, bivariate visualization merupakan bentuk visualisasi data untuk menyajikan informasi dari dua variabel dalam suatu dataset. Di lain sisi, multivariate visualization merupakan jenis visualisasi data untuk menggambarkan informasi yang terdapat dalam lebih dari dua variabel. Keduanya sering digunakan untuk mendeskripsikan hubungan antar variabel dalam suatu dataset.

#### Scatter Plot

Scatter plot merupakan bentuk visualisasi data yang digunakan untuk melihat hubungan antara dua atau lebih variabel data kuantitatif. Pada scatter plot sumbu X, Y, atau Z digunakan untuk menampung nilai dari setiap variabel yang akan divisualkan. Titik pada scatter plot merupakan titik temu data dari setiap sumbu yang digunakan.

Untuk membuat scatter plot, kita bisa menggunakan library matplotlib. Ia telah menyediakan sebuah function bernama **scatter()** untuk membantu kita membuat scatter plot secara lebih mudah.

Selain menggunakan library matplotlib, kita juga bisa menggunakan function **scatterplot()** yang disediakan oleh library seaborn untuk membuat scatter plot.

Untuk mempermudah dalam melihat korelasi atau hubungan antar variabel, kita bisa menggunakan function **regplot()** yang disediakan oleh library seaborn. Function tersebut memadukan scatter plot dan regression function (metode statistik untuk memperkirakan korelasi antar independent dan dependent variable) untuk melihat tren serta korelasi antar variabel.

#### Line Chart

Line chart merupakan bentuk grafik yang umum digunakan untuk menggambarkan tren nilai dari suatu variabel terhadap variabel lain. Kita bisa membuat line chart menggunakan function **plot()** yang disediakan oleh matplotlib.

Bentuk visualisasi data ini umumnya digunakan untuk melihat trend dari data berbentuk time series (data yang direkam dalam interval waktu yang konsisten).

#### Clustered Bar Chart

Clustered bar chart merupakan bentuk modifikasi dari bar chart yang sebelumnya kita kenal dengan menambahkan variabel kategoris lain. Modifikasi ini memungkinkan untuk melihat distribusi serta hubungan antar dua atau lebih variabel kategoris.

Untuk membuat clustered bar chart, kita bisa menggunakan parameter **hue** pada function **barplot()** yang disediakan oleh library seaborn.

#### Menggunakan Komponen Visual Tambahan

Selain menggunakan scatter plot, line chart, dan clustered bar chart, kita juga bisa memodifikasi bentuk visualisasi data dengan menambahkan komponen visual tambahan seperti color, shape, serta texture untuk mendeskripsikan lebih banyak variabel.

Sebagai contoh, Anda bisa membuat lebih dari satu boxplot untuk menggambarkan distribusi nilai numerik dari beberapa variabel. Hal ini bisa Anda gunakan dengan memasukkan DataFrame yang akan divisualisasikan pada parameter **data**dalam function **boxplot()**. Selain itu, Anda juga bisa menggunakan parameter **palette** untuk mengatur palet warna yang akan digunakan.

### Explanatory Analysis

Explanatory analysis merupakan proses analisis data yang bertujuan untuk membagikan beberapa insight yang menarik dari sebuah data. Insight yang diperoleh dari proses explanatory analysis akan digunakan untuk menjawab permasalahan bisnis yang dihadapi. Insight tersebut juga akan dibagikan kepada orang lain (stakeholder) sehingga perlu disertakan dengan visualisasi data yang baik

Pada proses explanatory analysis, visualisasi data digunakan sebagai media untuk menyampaikan pesan atau story dari sebuah data. Oleh karena itu, kita harus menyampaikan informasi terkait konteks dari sebuah visualisasi data. Hal ini bisa dicapai dengan menambahkan beberapa komponen berikut.

* **Title**  
  Title atau judul merupakan salah satu komponen penting guna menyampaikan konteks pada sebuah visualisasi data. Kita bisa menambahkan judul pada visualisasi data menggunakan function **title()** yang disediakan oleh matplotlib.
* **Label axes**  
  Selain judul visualisasi, kita juga perlu menambahkan label atau keterangan pada setiap sumbu yang terdapat dalam visualisasi data. Hal ini dilakukan untuk memberikan konteks terkait titik data yang direpresentasikan. Kita bisa membuat label axes menggunakan function **xlabel()** dan **ylabel()** yang disediakan oleh library matplotlib.
* **Legend**  
  Pastikan Anda menambahkan legend memberikan keterangan untuk variabel yang tidak digambarkan pada sumbu. Keterangan yang digunakan akan bergantung pada cara kita dalam merepresentasikan variabel tambahan tersebut. Sebagai contoh, kita bisa menggunakan **legend()** untuk memberikan keterangan pada variabel kategoris tambahan. Namun, apabila variabel tambahan tersebut berupa nilai numerik yang direpresentasikan menggunakan range warna, kita bisa memanfaatkan function **colorbar()** untuk memberikan keterangan pada variabel tersebut.

Selain informasi konteks, pastikan audiens fokus pada informasi yang ingin kita sampaikan. Oleh karena itu, sebagai seorang praktisi data yang andal, kita harus mampu mengarahkan perhatian audiens. Untuk melakukan hal ini, kita bisa menggunakan komponen visual tertentu seperti color, shape, dan texture.

Sebagai contoh, kita bisa menggunakan warna yang kontras untuk mengarahkan perhatian audiens pada bagian yang ingin di-highlight.

Namun, perlu diingat bahwa ketika menggunakan teknik ini kita harus menghindari penggunaan warna yang menyakitkan mata audiens atau warna yang memiliki arti berlawanan dengan pesan yang ingin disampaikan. Selain itu, kita juga memperhatikan konsistensi dalam pemilihan warna dalam seluruh visualisasi data yang kita buat.

Pada proses pembuatan explanatory analysis, terkadang sulit untuk menilai seberapa efektif visualisasi data yang kita buat. Oleh karena itu, kita membutuhkan feedback serta sudut pandang orang lain guna memastikan bahwa visualisasi data yang kita buat sudah cukup efektif dalam menyampaikan story dan informasi penting dari sebuah data.

Pada proses ini kita bisa meminta teman atau rekan setim untuk melihat seluruh visualisasi data yang telah kita buat. Selanjutnya, mintalah mereka untuk membuat kesimpulan dari seluruh visualisasi data yang telah dibuat. Lalu, pastikan kesimpulan mereka sesuai dengan yang ingin Anda sampaikan. Pada prosesnya, Anda bisa mengajukan beberapa pertanyaan berikut.

* Informasi apa yang Anda peroleh dari setiap visualisasi data tersebut?
* Menurut Anda apa kesimpulan utama dari seluruh visualisasi data tersebut?
* Apakah Anda memiliki pertanyaan terkait data tersebut yang belum terjawab?
* Apakah ada bagian yang belum Anda pahami dari seluruh visualisasi data tersebut?
* Berapa lama waktu yang Anda butuhkan untuk memahami setiap visualisasi data tersebut?

Anda dapat menggunakan beberapa pertanyaan tersebut guna mendapatkan feedback atau masukan untuk meningkatkan kualitas visualisasi data yang dibuat. Selain itu, pertanyaan tersebut juga membantu Anda mengidentifikasi informasi penting lain yang mungkin Anda lewatkan.

## Rangkuman Pengembangan Dashboard

### Pengenalan Dashboard

Saat ini, dashboard banyak digunakan sebagai media untuk memonitori berbagai metrik yang dianggap penting untuk perkembangan sebuah perusahaan. Selain itu, ia juga merupakan salah satu komponen penting dalam kultur data-driven decision making. Umumnya, sebuah dashboard terdiri dari dari beberapa visualisasi data yang interaktif dan saling berhubungan satu sama lain.

Dashboard yang efektif merupakan kunci utama dalam kultur data-driven decision making. Berikut merupakan beberapa hal yang harus diperhatikan dalam membuat dashboard yang efektif.

* **Memahami audiens**  
  Untuk membuat dashboard yang efektif, kita harus mempertimbangkan audiens dan kebutuhannya. Oleh karena itu, pada proses pembuatan dashboard kita harus memahami siapa target user atau audiens serta goals atau kebutuhan yang ingin mereka capai dari dashboard tersebut.
* **Pertimbangkan ukuran tampilan**  
  Sejalan dengan poin pertama, kita juga harus mempertimbangkan ukuran tampilan dashboard yang sesuai dengan device atau perangkat yang sering digunakan oleh target user atau audiens untuk melihat dashboard yang Anda buat.
* **Manfaatkan sweet spot**  
  Anda bisa menggunakan sweet spot untuk menampilkan informasi penting dari sebuah dashboard. Tentunya Anda perlu melakukan riset terlebih dahulu untuk mengetahui sweet spot yang tepat untuk kebutuhan Anda.
* **Pertimbangkan load time**  
  Load time merupakan salah hal yang penting untuk kita pertimbangkan guna menjaga pengalaman user atau audiens. Oleh karena itu, kita harus memastikan dashboard yang dibuat memiliki load time yang relatif singkat.
* **Batasi komponen visual yang ditampilkan**  
  Pada proses pengembangan dashboard kita juga harus mempertimbangkan jumlah komponen visual yang akan ditampilkan. Secara umum, kita disarankan untuk hanya menampilkan dua atau tiga visualisasi data serta menghindari penggunaan komponen visual yang tidak mendukung dalam penyampaian informasi penting.

### Pengenalan Streamlit

Streamlit merupakan salah satu open-source web app framework untuk bahasa pemrograman Python. Ia memungkinkan kita membuat web app yang baik dan interaktif dalam waktu yang singkat. Selain itu, ia juga kompatibel dengan berbagai library populer seperti NumPy, pandas, matplotlib, dll. Inilah yang menjadi alasan streamlit sering digunakan oleh oleh para praktisi data dan machine learning.

Seperti biasa, sebelum mulai menggunakan streamlit tentunya kita perlu menyiapkan environment-nya terlebih dahulu. Berikut merupakan tahapan untuk melakukannya.

1. Untuk menggunakan streamlit, kita perlu menulis kode Python dalam berkas**.py**. Oleh karena itu, Anda perlu menginstal code editor, seperti [Visual Studio Code](https://code.visualstudio.com/download), [Sublime Text](https://www.sublimetext.com/3), atau [PyCharm](https://www.jetbrains.com/pycharm/download/#section=windows) untuk mempermudah dalam menulis kode.
2. Selanjutnya, Anda perlu mengaktifkan virtual environment sebelumnya yang telah kita gunakan dalam materi latihan. Aktifkan virtual environment dengan menjalankan perintah berikut.
   1. conda activate main-ds
3. Tahap berikutnya adalah menginstal streamlit. Berikut merupakan perintah yang bisa Anda gunakan untuk menginstal streamlit.
   1. pip install streamlit

### Basic Element dalam Streamlit

Sebagai sebuah web app framework yang andal, streamlit telah menyediakan banyak pilihan element, widget, layout serta container untuk memastikan kita dapat membuat web app atau dashboard yang menarik dan interaktif.

#### Write

Basic element pertama yang akan kita bahas ialah write. Ia merupakan elemen streamlit yang digunakan untuk menampilkan sebuah output.  Untuk menggunakan element ini, kita hanya perlu memanggil function **write()** dan diikuti inputan yang ingin ditampilkan.

#### Text

Elemen lain yang ada dalam streamlit ialah text. Sesuai dengan namanya, ia merupakan elemen yang digunakan untuk menampilkan sebuah output berupa text. Elemen ini memiliki banyak function yang bisa digunakan sesuai kebutuhan. Berikut merupakan beberapa pilihan yang tersedia saat ini.

* **markdown()**  
  Function ini digunakan untuk menampilkan output dari argument markdown.
* **title()**  
  Ini merupakan function yang digunakan untuk menampilkan teks dalam format judul.
* **header()**  
  Function ini digunakan untuk menampilkan output teks sebagai format header.
* **subheader()**  
  Ini merupakan function yang digunakan untuk menampilkan output teks sebagai format subheader.
* **caption()**  
  Ia merupakan function yang digunakan untuk menampilkan output teks dalam ukuran kecil. Function ini biasanya digunakan untuk menampilkan caption, footnotes, dll.
* **code()**  
  Pada beberapa case, kita harus menampilkan potongan kode ke dalam streamlit app (web app yang dibuat menggunakan streamlit). Untuk menjawab hal ini, streamlit telah menyediakan sebuah function bernama **code()**.
* **text()**  
  Function selanjutnya ialah **text()**. Function ini digunakan untuk menampilkan sebuah normal teks.
* **latex()**  
  Function terakhir yang dapat digunakan untuk menampilkan elemen teks ialah latex(). Sesuai namanya, function tersebut digunakan untuk menampilkan mathematical expression yang ditulis dalam [format LaTeX](https://katex.org/docs/supported.html).

#### Data Display

Basic element selanjutnya yang akan kita bahas ialah data display. Ia merupakan elemen yang digunakan untuk menampilkan data secara cepat dan interaktif ke dalam streamlit app yang kita buat. Elemen ini memiliki beberapa function seperti berikut.

* **dataframe()**  
  Ia merupakan function yang digunakan untuk menampilkan DataFrame sebagai sebuah tabel interaktif. Pada function ini, kita bisa mengatur ukuran dari table yang ingin ditampilkan menggunakan parameter **width** dan **height**.
* **table()**  
  Selain **dataframe()**, kita juga bisa menggunakan function **table()** untuk menampilkan data ke dalam streamlit app. Ia dapat digunakan untuk menampilkan data dalam bentuk static table.
* **metric()**  
  Ketika membuat dashboard, terkadang kita perlu menampilkan sebuah metrik tertentu. Untuk melakukan hal ini, kita bisa memanfaatkan function **metric()**.
* **json()**  
  Selain bentuk DataFrame atau tabel, terkadang kita juga perlu menampilkan data dalam format JSON. Streamlit telah menyediakan function json() untuk menampilkan data dalam format JSON.

#### Chart

Basic element terakhir yang perlu kita ketahui ialah chart. Sesuai namanya, elemen ini dapat digunakan untuk menampilkan grafik visualisasi data ke dalam streamlit app. Elemen inilah yang akan sering kita gunakan untuk membuat dashboard nantinya.

Function **pyplot()** dapat digunakan untuk menampilkan grafik visualisasi data yang dibuat menggunakan matplotlib.

### Basic Widgets dalam Streamlit

Untuk menghasilkan sebuah dashboard yang interaktif, kita memerlukan komponen lain seperti widget (elemen Graphical User Interface yang memungkinkan pengguna untuk berinteraksi dengan aplikasi).

#### Input Widget

Kategori widget yang akan kita bahas ialah input widget. Ia merupakan kategori widget yang memungkinkan pengguna untuk memberikan input ke dalam streamlit app. Terdapat beberapa widget yang termasuk kategori ini, seperti text input, number input, date input, dll. Yuk, kita bahas satu per satu!

* **Text input**  
  Text input merupakan widget yang digunakan untuk memperoleh inputan berupa single-line text. Kita bisa menggunakan function **text\_input()** untuk membuat widget ini.
* **Text-area**  
  Text area merupakan widget yang memungkinkan pengguna untuk menginput multi-line text. Untuk membuat widget ini, streamlit telah menyediakan function bernama **text\_area()**.
* **Number input**  
  Ia merupakan widget yang digunakan untuk memperoleh inputan berupa angka dari pengguna. Anda dapat menggunakan function **number\_input()** untuk membuat number input widget.
* **Date input**  
  Selain inputan berupa angka dan teks, pada beberapa kasus kita juga membutuhkan input berupa tanggal dari pengguna melalui date input widget. Kita dapat membuat widget tersebut menggunakan function **date\_input()**.
* **File uploader**  
  Widget ini memungkinkan kita meminta pengguna untuk meng-upload sebuah berkas tertentu ke dalam web app. Kita dapat membuat file uploader widget menggunakan function  **file\_uploader()**.
* **Camera input**  
  Selain file uploader, streamlit juga menyediakan camera input widget yang dapat digunakan untuk meminta user mengambil gambar melalui webcam sekaligus mengunggahnya menggunakan function**camera\_input()**.

#### Button Widgets

Oke, kategori widget selanjutnya yang akan kita bahas ialah button widget. Ia merupakan kategori widget yang terdiri dari button, checkbox, radio button, dll.

* **Button**  
  Button merupakan widget untuk menampilkan tombol interaktif. Tombol tersebut dapat digunakan pengguna untuk melakukan aksi tertentu. Untuk membuat widget ini, kita bisa menggunakan function **button()**.
* **Checkbox**  
  Checkbox merupakan widget yang digunakan untuk menampilkan sebuah checklist untuk pengguna. Kita bisa menggunakan function **checkbox()** untuk membuat dan menampilkan checklist dalam streamlit app.
* **Radio button**  
  Ia merupakan widget yang memungkinkan pengguna untuk memilih satu dari beberapa pilihan yang ada. Untuk membuat widget ini, kita bisa menggunakan function **radio()**.
* **Select Box**  
  Select box merupakan widget yang memungkinkan pengguna untuk memilih salah satu dari beberapa pilihan yang ada. Ia merupakan opsi alternatif dari radio button. Streamlit telah menyediakan function **selectbox()** untuk membuat select box widget.
* **Multiselect**  
  Ia merupakan widget yang digunakan agar user dapat memilih lebih dari satu pilihan dari sekumpulan pilihan yang ada. Untuk mempermudah kita dalam membuat multiselect widget, streamlit telah menyediakan function bernama **multiselect()**.
* **Slider**  
  Slider merupakan widget yang memungkinkan pengguna untuk untuk memilih nilai (atau range nilai) dari sebuah range nilai yang telah ditentukan. Streamlit telah menyediakan function **slider()** untuk membuat slider widget.

### Basic Layouts dalam Streamlit

Untuk membuat dashboard yang rapi, kita perlu belajar cara mengatur layout atau tampilan dari sebuah dashboard. Sebagai sebuah web app framework, streamlit telah menyediakan berbagai pilihan layout yang dapat digunakan untuk mengatur tampilan web app (atau dashboard) yang akan dibuat.

#### Sidebar

Elemen layout pertama yang akan kita bahas ialah sidebar. Ia merupakan area yang berada di samping konten utama. Pada streamlit, posisi sidebar secara default berada di bagian kiri dari konten utama dan dapat memuat berbagai hal mulai dari gambar, teks, hingga widget.

Untuk menambahkan sebuah elemen atau widget ke dalam sidebar, kita bisa menggunakan notasi **with** yang diikuti sebuah object bernama **sidebar** yang telah disediakan oleh streamlit.

#### Columns

Columns merupakan elemen layout yang memungkinkan kita untuk mengatur tampilan pada konten utama ke dalam beberapa kolom sesuai kebutuhan.

Untuk membuat column dalam streamlit app, kita harus membuat object dari setiap kolom terlebih dahulu. Hal ini dapat dilakukan dengan memanfaatkan function **columns()**. Selanjutnya, kita hanya perlu menambahkan sebuah elemen atau widget ke dalam column tersebut menggunakan notasi **with**.

#### Tabs

Ia merupakan elemen layout yang memungkinkan kita untuk menambahkan beberapa tabs ke dalam konten utama. Hal ini tentunya akan sangat membantu kita dalam menghasilkan dashboard yang interaktif.

Sama halnya dengan columns yang sebelumnya kita bahas, untuk membuat tabs, kita harus membuat object dari setiap tab menggunakan function **tabs()** yang diikuti dengan list dari nama dari setiap tab.

#### Container

Container merupakan elemen layout dalam streamlit yang memungkinkan kita untuk membuat sebuah wadah untuk menampung suatu atau beberapa elemen dengan ukuran yang tetap. Container ini dapat kita gunakan untuk menghasilkan dashboard yang rapi.

Untuk membuat container, kita bisa menggunakan notasi **with** dan diikuti function **container()**.

#### Expander

Elemen layout selanjutnya yang tidak kalah penting ialah expander. Anda dapat menganggap elemen layout ini sebagai sebuah container yang dapat diperluas atau diciutkan secara interaktif.

Untuk membuat elemen layout ini, kita bisa menggunakan notasi **with** dan diikuti dengan function **expander()**.