**Pengenalan Exploratory Data Analysis**

Seperti yang telah kita bahas pada materi awal, 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.

Berdasarkan sudut pandang para praktisi data, proses EDA merupakan salah satu tahapan yang paling menarik dalam proyek analisis data. Proses ini memungkinkaan para praktisi data untuk melakukan eksplorasi dan bereksperimen terhadap sebuah data guna mencari pola, memperoleh insight, menjawab berbagai tantangan bisnis, serta membuat sebuah kesimpulan berdasarkan hasil analisis.

Sebagai calon praktisi data masa depan, pastinya Anda akan menikmati proses EDA ini. Sebagai gambaran, pada beberapa materi ke depan, kita akan membahas berbagai tool dan teknik yang dapat membantu Anda dalam melakukan proses EDA secara efektif. Namun, sebelum membahas semua hal tersebut, alangkah baiknya jika kita memahami perbedaan antara *exploratory analysis* dan *explanatory analysis* terlebih dahulu.

**Exploratory Analysis vs Explanatory Analysis**

Exploratory analysis dan explanatory analysis merupakan dua istilah sering dijumpai dalam proyek analisis data. Selain pelafalannya yang cukup mirip, kedua istilah tersebut juga mewakili proses yang cukup mirip. Hal inilah yang membuat para praktisi data pemula sulit untuk membedakan kedua istilah tersebut.

Sederhana, exploratory analysis merupakan proses analisis data yang bertujuan untuk mengeksplorasi dan mengenal sebuah data [10]. Pada prosesnya sering kali mulai dengan mendefinisikan berbagai pertanyaan atau hanya menggali data guna menemukan beberapa insight yang menarik dari sebuah data. Pada proses ini, terkadang kita juga menerapkan visualisasi data sederhana yang bertujuan untuk memvalidasi insight yang diperoleh. Jika dianalogikan, proses exploratory analysis seperti proses penambangan yang mengumpulkan 100 batu untuk menemukan satu atau dua buah berlian yang berharga.



Di lain sisi, explanatory analysis merupakan proses analisis data yang bertujuan untuk membagikan beberapa insight yang menarik dari sebuah data (bisa diibaratkan sebagai satu atau dua buah berlian yang berharga) [10]. 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.

Nah, sekarang Anda telah memahami perbedaan antara exploratory analysis dan explanatory analysis. Pada beberapa materi ke depan, kita akan mulai dengan membahas berbagai teknik dalam exploratory analysis.

**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. Pertanyaan tersebut akan menjadi kompas bagi kita pada setiap tahap dalam proses analisis data.

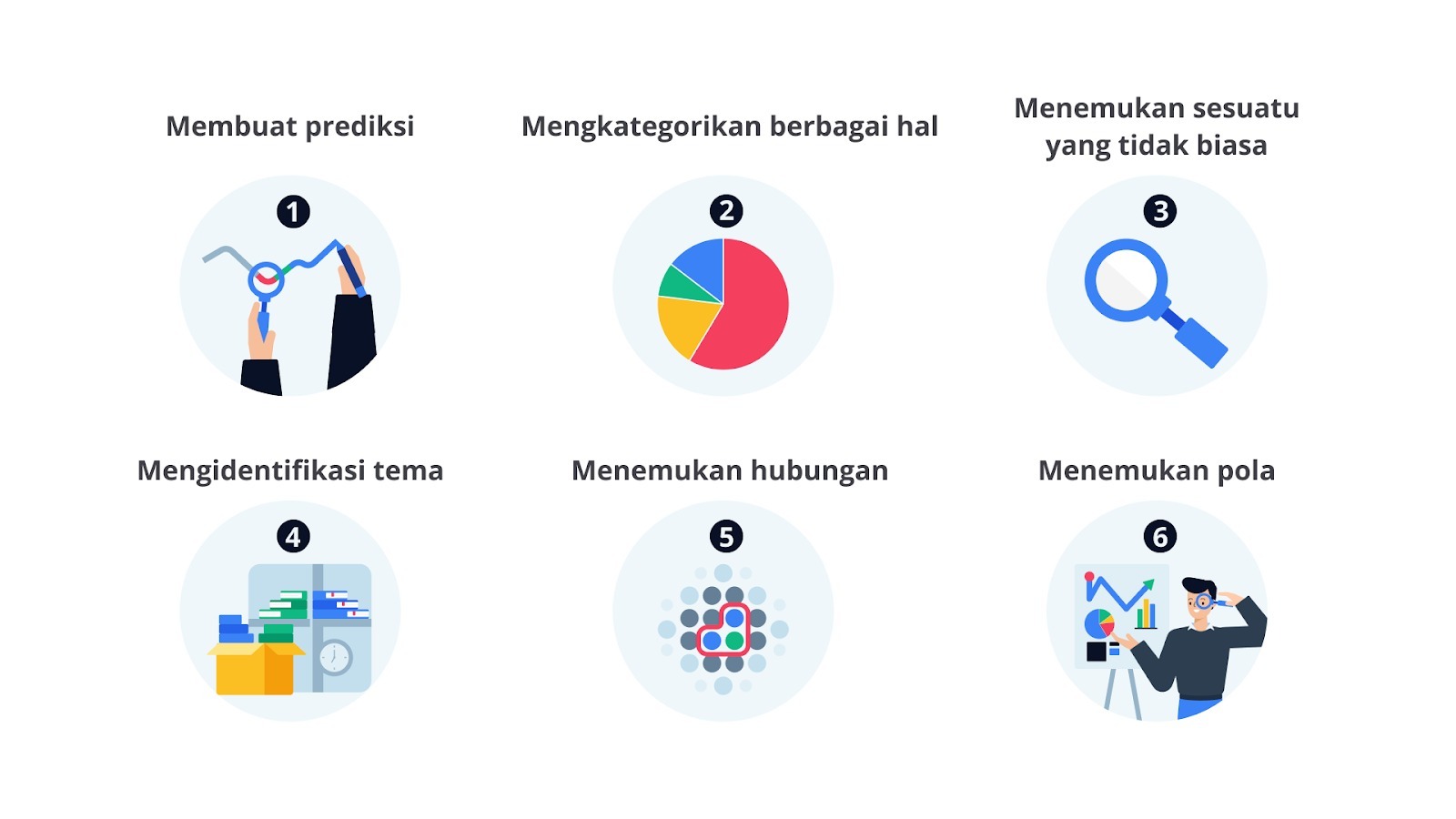
Nah, pada materi kali ini, kita akan belajar cara membuat pertanyaan yang baik dan efektif untuk memperoleh *meaningful insight*. Hal ini tentunya dapat membantu kita dalam memahami sebuah permasalahan bisnis dan ekspektasi dari para pemangku kepentingan (*stakeholder*). Sebagai permulaan, kita akan awali materi ini dengan membahas tentang *problem solving*.

**Problem Solving**

Problem solving merupakan salah satu keterampilan penting yang harus dimiliki seorang praktisi data. Sebagai calon praktisi data masa depan, Anda harus mampu melihat sebuah problem sebagai peluang untuk memanfaatkan keterampilan yang dimiliki dalam mencari sebuah solusi.

Untuk menyelesaikan sebuah problem, kita harus memahami terlebih dahulu permasalahan yang ingin diselesaikan.. Secara umum, terdapat enam kategori problem yang sering dijumpai oleh para praktisi data di industri. Berikut merupakan keenam kategori tersebut [11].

* **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.  
    
  Sebagai contoh, Anda diminta untuk mengidentifikasi strategi *advertising* yang paling optimal untuk meningkatkan jumlah pengguna dari sebuah platform. Pada masalah ini, Anda harus menggunakan data sebelumnya untuk melihat performa dari setiap pilihan advertising yang ada.
* **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.  
    
  Sebagai contoh, sebuah perusahaan ingin mengenal customer secara lebih mendalam. Untuk menyelesaikan masalah ini, Anda harus menganalisis data histori aktivitas customer. Data tersebut digunakan untuk mengidentifikasi kesamaan yang ada dari setiap customer. Kesamaan inilah yang akan digunakan untuk mengategorikan customer ke dalam beberapa kluster.
* **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.  
    
  Sebagai contoh, Anda diminta untuk mencari tahu penyebab penurunan jumlah pengguna dalam sebuah platform. Untuk menyelesaikan problem ini, Anda harus menganalisis tingkat kepuasan pengguna terhadap layanan dan UI/UX yang dihadirkan oleh platform tersebut. Selain itu, Anda juga harus mempertimbangkan hal lain seperti perubahan perilaku pada *targeted user*, keberadaan kompetitor, dan berbagai pengaruh lainnya.
* **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 pengalaman 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.  
    
  Sebagai contoh, sebuah perusahaan *e-commerce* ingin meningkatkan efisiensi dalam proses pengiriman barang. Untuk melakukan hal tersebut, seorang praktisi data harus menganalisis hubungan antara data order dan *waiting time* pada setiap fase pengiriman barang. Hasil dari analisis tersebut dapat membantu dalam menentukan jadwal pengiriman barang paling sesuai untuk meningkatkan pengiriman barang tepat waktu.
* **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.  
    
  Contohnya, sebuah perusahaan yang memiliki layanan *streaming*ingin meningkatkan jumlah film atau series yang ditonton pengguna. Untuk melakukan hal ini, kita dapat menganalisis data histori pengguna untuk menemukan pola yang terdapat di dalamnya. Pola ini dapat dijadikan acuan dalam memberikan rekomendasi film atau series untuk di tonton pengguna lain.



Oke, itulah keenam kategori problem yang umum dijumpai oleh para praktisi data di industri. Anda dapat menggunakan kategori tersebut untuk memahami masalah bisnis yang dihadapi.

Nah, setelah memahami masalah bisnis yang dihadapi, tahap berikutnya ialah membuat perencanaan untuk menyelesaikan masalah tersebut. Untuk melakukannya, para praktisi data selalu mulai dengan mendefinisikan beberapa pertanyaan yang tepat dan efektif.

**Membuat Pertanyaan yang Efektif**

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. Contoh: tampilan UI produk ini menarik, bukan?
* Hindari membuat pertanyaan yang hanya dapat dijawab dengan “ya” atau “tidak.” Contoh: apakah user menyukai tampilan UI baru?
* Hindari membuat pertanyaan yang tidak memiliki konteks. Contoh: user lebih suka versi 1 atau versi 2?

Untuk menghasilkan pertanyaan analisis yang efektif, terdapat beberapa poin yang harus Anda perhatikan seperti berikut [11].

* **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. Oleh karena itu, jika sebuah pertanyaan masih bersifat general, cobalah mempersempitnya dengan fokus pada satu topik tertentu.
* **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. Sebagai contoh, ketimbang bertanya “Apakah user menyukai tampilan UI baru?” Kita bisa mengubah pertanyaan tersebut menjadi “Berapa persen user yang menyukai tampilan UI baru dari produk yang kita buat?”
* **Action-oriented**  
  Pertanyaan yang bersifat *action-oriented* berarti pertanyaan tersebut mampu memberikan informasi yang dapat membantu kita dalam membuat sebuah action-plan. Contoh: design UI seperti apa yang dapat memperbaiki pengalamanan user ketika berinteraksi dengan produk yang kita miliki?
* **Relevant**  
  Pertanyaan yang relevan berarti pertanyaan tersebut berbobot dan memiliki arti untuk menyelesaikan sebuah permasalahan. Sebagai contoh, Anda diminta untuk mengidentifikasi strategi advertising yang paling optimal untuk meningkatkan jumlah pengguna dari sebuah produk. Untuk menyelesaikan permasalahan ini, Anda dapat membuat pertanyaan, “Bagaimana performa dari setiap pilihan advertising yang ada terhadap perkembangan produk yang kita miliki?”
* **Time-bound**  
  Terakhir, pertanyaan yang efektif haruslah bersifat time-bound yang berarti pertanyaan tersebut haruslah memiliki relevansi terhadap waktu saat ini. Sebagai contoh, jika Anda sedang menganalisis data harga aset saham perusahaan A, pertanyaan yang tepat untuk kasus ini ialah “Bagaimana performa harga aset saham perusahaan A dari tahun 2005 hingga 2022?”

Oke, itulah lima poin yang harus Anda perhatikan ketika membuat sebuah pertanyaan analisis. Semoga pembahasan ini dapat membantu dalam membuat pertanyaan yang baik dan efektif sehingga dapat mempermudah Anda dalam mengerjakan sebuah proyek analisis data.

Setelah memahami permasalahan bisnis yang dihadapi dengan cara membuat pertanyaan analisis yang efektif, kini saatnya Anda melihat berbagai tool dan teknik yang dapat membantu Anda dalam melakukan Exploratory Data Analysis. Apakah Anda sudah siap? Yuk, kita pindah ke materi berikutnya!

**Mengeksplorasi Parameter Statistik dari Data**

Sampai dengan tahap ini, Anda telah memahami berbagai hal dalam merumuskan pertanyaan analisis yang efektif. Pekerjaan seorang praktisi data tentunya tidak berhenti sampai di situ saja, melainkan ia juga harus mencari jawaban dari pertanyaan tersebut.

Proses pencarian jawaban tersebut tentunya tidak akan mudah dan penuh dengan tantangan. Namun, tenang jangan risau karena kelas ini akan membantu Anda dalam memahami proses EDA guna menemukan jawaban dari pertanyaan yang sudah dibuat sebelumnya.

Sebagai permulaan, kita akan belajar berbagai tool yang dapat membantu kita dalam mengeksplorasi parameter statistik secara lebih efisien dan *scalable*. Apakah Anda sudah siap? Mari kita mulai!

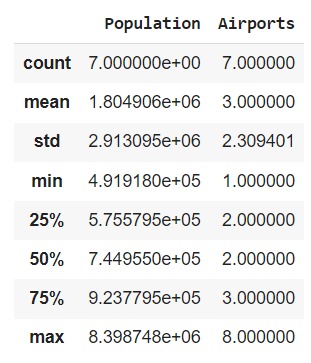
*So*, pada materi Penerapan Dasar-Dasar Descriptive Statistics, kita mempelajari berbagai parameter statistik yang umum digunakan dalam mendeskripsikan sebuah data. Tentunya Anda masih ingat dengan seluruh parameter tersebut, bukan? Nah, pada materi kali ini, kita akan menggunakan berbagai parameter tersebut untuk memahami sebuah data.

“*Hm…,* berarti kita harus memeriksa parameter tersebut satu per satu, ya?” Jika ini yang terlintas di benak Anda, jangan khawatir karena *library* tercinta kita yang bernama pandas telah menyediakan berbagai *method* yang bisa memudahkan proses identifikasi parameter statistik dari sebuah data.

Method pertama yang akan kita bahas ialah method describe(). Method ini memungkinkan kita untuk melihat rangkuman parameter statistik (mean, median, dll.) dari kolom numerik pada sebuah DataFrame. Berikut contoh kode untuk menggunakan method describe().

|  |
| --- |
| import pandas as pd    city\_names = ['Jakarta', 'Bandung', 'Makassar', 'Surabaya', 'Medan', 'Yogyakarta', 'Malang']  population = [498044, 964254, 491918, 8398748, 653115, 883305, 744955]  num\_airports = [2, 2, 8, 3, 1, 3, 2]    df = pd.DataFrame({  'City Name': city\_names,  'Population': population,  'Airports': num\_airports,  })    df.describe() |

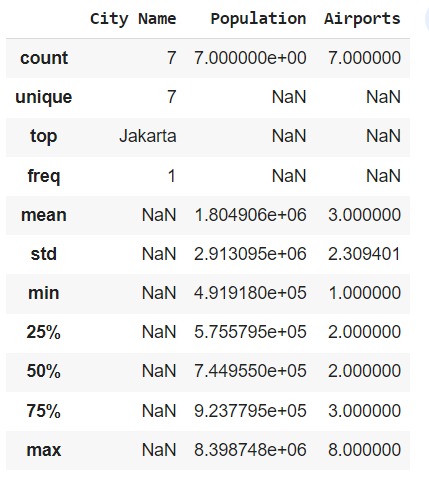
Rangkuman parameter statistik yang diperoleh ketika Anda menjalankan kode di atas.



Jika ingin melihat lebih banyak parameter statistik serta menyertakan kolom non-numerik, Anda bisa menggunakan parameter include=”all” seperti contoh berikut

|  |
| --- |
| df.describe(include="all") |

Dengan menerapkan parameter tersebut, Anda akan menemukan tampilan rangkuman parameter statistik seperti berikut.

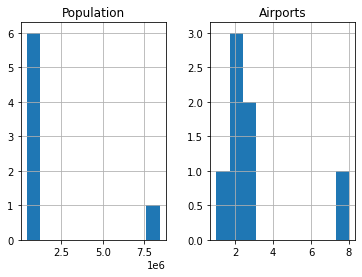


Nah, jika Anda perhatikan, terdapat beberapa parameter yang bernilai NaN. Hal ini menunjukkan bahwa kita tidak dapat menghitung nilainya pada kolom tersebut. Sebagai contoh, kolom **Population** tidak memiliki nilai unique karena ia merupakan kolom dengan tipe data kontinu. Di sisi lain, kita tidak dapat menghitung nilai mean pada kolom **City Name** karena ia merupakan kolom kategorik.

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. Berikut contoh kode untuk menggunakan method hist().

|  |
| --- |
| df.hist() |

Kode di atas akan menghasilkan plot histogram seperti di bawah ini.



Berdasarkan plot tersebut, dapat dilihat bahwa data pada kolom Populasi dan Airports memiliki distribusi *right-skewed*.

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

Nilai korelasi yang diperoleh dari kode di atas adalah sebagai berikut.



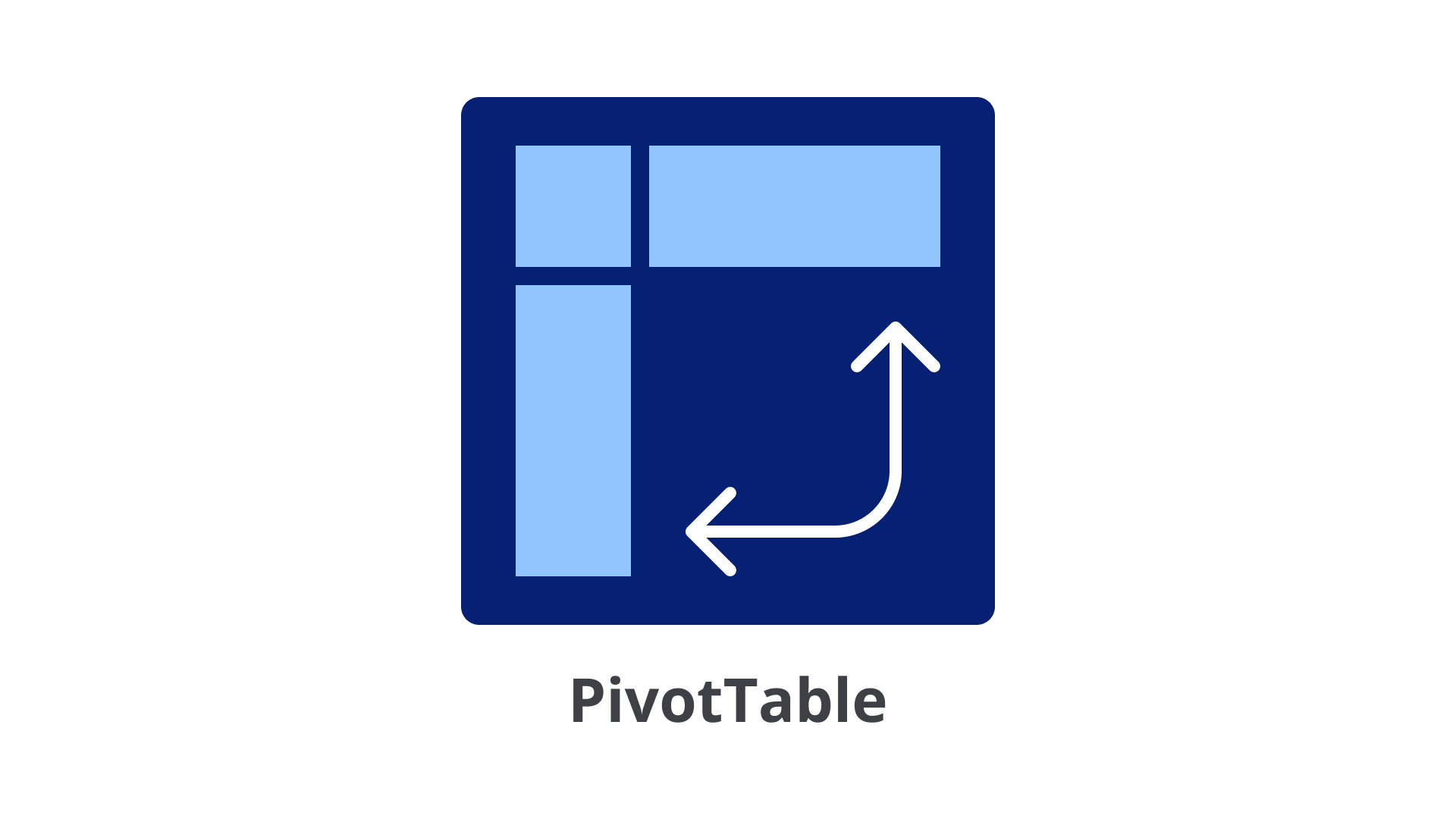
Berdasarkan nilai korelasi tersebut, kita dapat menyimpulkan bahwa kedua kolom di atas tidak memiliki korelasi karena nilainya mendekati nol.

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. Silakan kunjungi dokumentasi berikut untuk melihat lebih detail terkait hal tersebut: [dokumentasi pandas.DataFrame](https://pandas.pydata.org/docs/reference/frame.html" \t "_blank).

**Mengelompokkan Data**

Pada materi sebelumnya, kita telah mempelajari berbagai *method* dari library pandas yang dapat digunakan untuk mengeksplorasi parameter statistik dari sebuah data secara efisien dan *scalable*. Nah, untuk melengkapi *toolbox* Anda dalam mengeksplorasi data, pada materi kali ini akan dibahas berbagai hal terkait pengelompokan 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.



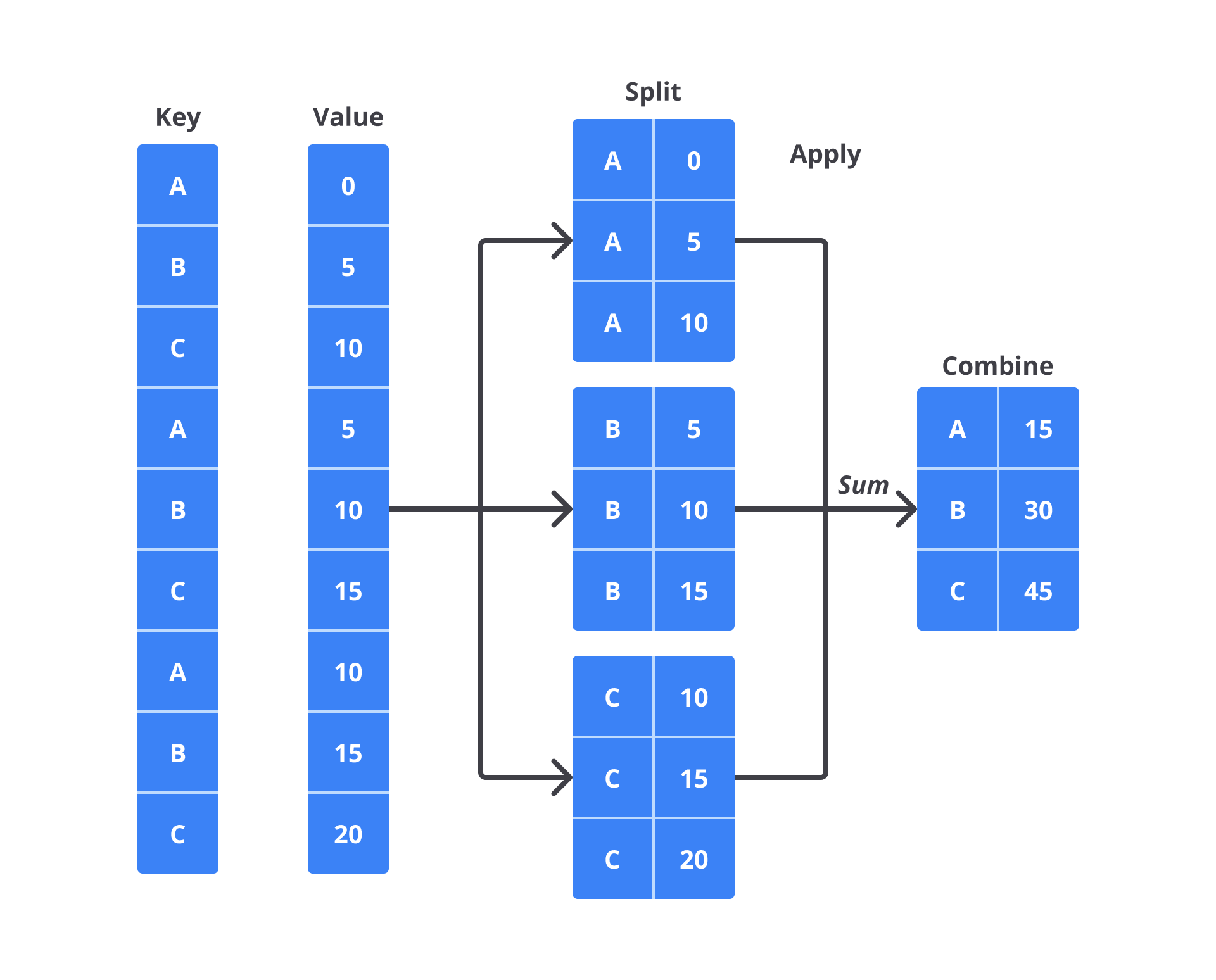
Pada proses analisis data, kita bisa menggunakan konsep pivot table ini untuk menghitung parameter statistik dari kelompok data tertentu. Selain itu, ia juga dapat digunakan untuk kebutuhan visualisasi data dan pembuatan *report*. Hal inilah yang membuat pengetahuan tentang pivot table menjadi sangat penting bagi para praktisi data. Namun, perlu diingat  bahwa di industri kita harus selalu mementingkan efisiensi dan *scalability* dari setiap tahapan analisis data yang dilakukan termasuk pembuatan pivot table. Oleh karena itu, pada materi kali ini, kita akan belajar cara membuat pivot table secara efisien dan scalable.

Akan tetapi, sebelum kita membahas hal tersebut, alangkah baiknya jika kita membahas prosesnya terlebih dahulu.

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.

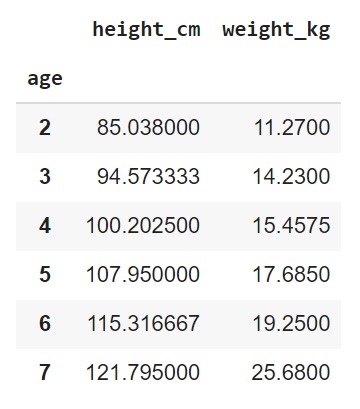


Nah, setelah mengetahui proses dalam pembuatan pivot table, sekarang saatnya kita belajar cara membuat pivot table secara efisien dan scalable. Seperti biasa, kita akan menggunakan library pandas untuk melakukannya.

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. Berikut contoh kode untuk menggunakan method groupby().

|  |
| --- |
| Import pandas as pd    body\_measurement\_df = pd.DataFrame.from\_records((  (2, 83.82, 8.4),  (7, 121.92, 22.96),  ), columns=("age", "height\_cm", "weight\_kg"))    body\_measurement\_df.groupby(by="age").mean() |

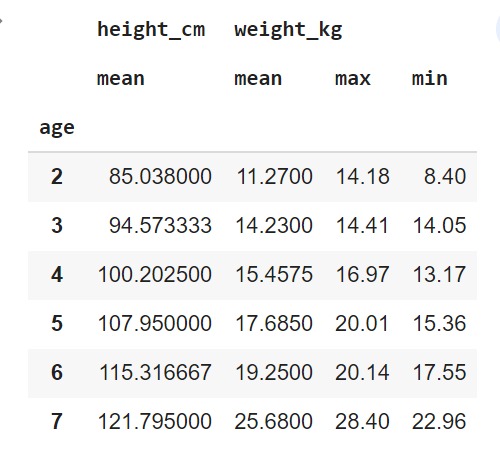
Berikut tampilan pivot table yang dihasilkan dari kode di atas.



Ketika melihat hasil di atas, mungkin Anda bertanya “bagaimana jika kita ingin membuat pivot table dengan lebih dari satu parameter?” Untuk menjawab tantangan ini, kita bisa menggunakan method agg(). Ia memungkinkan kita untuk menentukan beberapa parameter yang ingin digunakan dalam pivot table. Berikut merupakan contoh kode untuk menggunakan method agg().

|  |
| --- |
| body\_measurement\_df.groupby(by='age').agg({  'height\_cm': 'mean',  'weight\_kg': ['mean','max', 'min'],  }) |

Contoh kode tersebut akan menghasilkan pivot table seperti berikut.



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" \t "_blank).

Pada materi berikutnya, kita akan belajar menerapkan berbagai teori yang telah dipelajari sejauh ini melalui sebuah proyek latihan analisis sederhana. *So*, tanpa berlama-lama lagi. Mari kita lanjut ke materi berikutnya!