# [Exploratory Data Analysis with Python for Beginner](https://academy.dqlab.id/main/package/practice/163)

## CHAPTER 1

### Pendahuluan

Pagi penuh kejutan! Aku menemukan seseorang dengan rambut bergaya jamur dan bertubuh tambun menempel selembar memo berisi poin-poin materi seputar analisis data di dinding mejaku.

“Pelajari dalam waktu 3 hari” ujarnya singkat lalu pergi. Tantangan datang bertubi-tubi, inikah rasanya lika-liku meraih mimpi? Kutarik napas panjang, aku enggak akan nyerah!

Library pada Python merupakan kumpulan code yang bersifat open-source yang dapat dipanggil ke dalam Python dan digunakan untuk membantu komputasi. Library dasar pada Python yang digunakan untuk analisis data antara lain NumPy, SciPy, Pandas, dan Matplotlib dengan fungsional yang berbeda - beda. Modul ini akan menjabarkan kegunaan masing - masing library beserta contoh kasus penggunaan library tersebut dalam menyelesaikan real - case di dunia kerja.

### Library NumPy

Sejak hari pertama memasuki dunia data, aku sering kali menemukan istilah umum yang punya arti berbeda di bidang data. Contohnya nih, Python ternyata bukan nama jenis ular, tapi bahasa pemrograman. Lalu, sekarang aku menemukan modul pembelajaran seputar “Library dalam Python”! Kupikir sejenis perpustakaan, tapi tentunya bukan. Aku pernah menyampaikan hal ini pada Senja yang justru menertawaiku.

“Hahaha, dasar Aksara! Tidak apa-apa, perlahan kamu akan terbiasa. Setidaknya kamu jadi belajar hal baru bukan?”

“Betul,” ujarku bersemangat. Aku pun sudah tidak sabar mencari tahu apa itu “Library” versi Python ini. Ah, jadi kangen Senja! Hari ini aku harus belajar sendiri dengan materi-materi baru dari Andra karena memang untuk pembelajaran ini, Andra ahlinya.

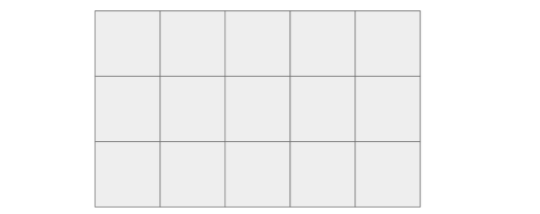
Aku pun mulai membolak-balik halaman modul baru yang kudapat tadi.

Numpy berasal dari kata ‘*Numerical Python’*, sesuai namanya NumPy berfungsi sebagai library untuk melakukan proses komputasi numerik terutama dalam bentuk *array* multidimensional (1-Dimensi ataupun 2-Dimensi). *Array* merupakan kumpulan dari variabel yang memiliki tipe data yang sama. NumPy menyimpan data dalam bentuk *arrays*.

Bentuk 1D NumPy *array*dapat diilustrasikan sebagai berikut:



Bentuk 2D NumPy *array* dapat diilustrasikan sebagai berikut:

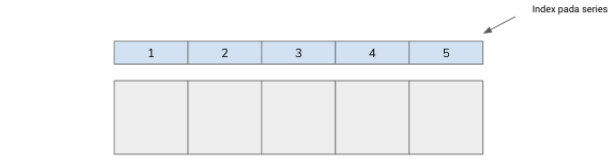


### Library Pandas

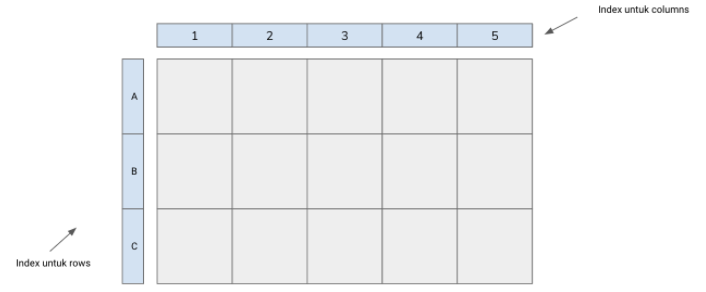
Pandas merupakan library yang memudahkan dalam melakukan manipulasi, *cleansing* maupun analisis struktur data. Dengan menggunakan Pandas, dapat memanfaatkan lima fitur utama dalam pemrosesan dan analisis data, yaitu *load*, *prepare*, *manipulate*, *modelling*, dan *analysis* data.

Pandas menggunakan konsep array dari NumPy namun memberikan index kepada array tersebut, sehingga disebut *series* ataupun *data frame*. Sehingga bisa dikatakan Pandas menyimpan data dalam *dictionary-based* NumPy arrays. 1-Dimensi labelled array dinamakan sebagai *Series*. Sedangkan 2-Dimensi dinamakan sebagai *Data Frame*.

Bentuk dari *series* dilustrasikan sebagai berikut



Bentuk dari *data frame* diilustrasikan sebagai berikut:



### Library SciPy

Scipy dibangun untuk bekerja dengan array NumPy dan menyediakan banyak komputasi numerik yang ramah pengguna dan efisien seperti rutinitas untuk integrasi, diferensiasi dan optimasi numerik.

Baik NumPy maupun SciPy berjalan pada semua operating system, cepat untuk diinstall dan gratis. NumPy dan SciPy mudah digunakan, tetapi cukup kuat untuk diandalkan oleh beberapa data *scientist* dan *researcher* terkemuka dunia.

### Library Matplotlib

Matplotlib merupakan library dari Python yang umum digunakan untuk visualisasi data. Matplotlib memiliki kapabilitas untuk membuat visualisasi data 2-dimensional. Contoh visualisasi yang dapat dibuat dengan menggunakan matplotlib diantaranya adalah

1. Line chart
2. Bar chart
3. Pie chart
4. Box plot chart
5. Violin chart
6. Errorbar chart
7. Scatter chart

Jenis-jenis *chart* lainnya juga dapat dibuat melalui library ini.

### Memanggil library di Python

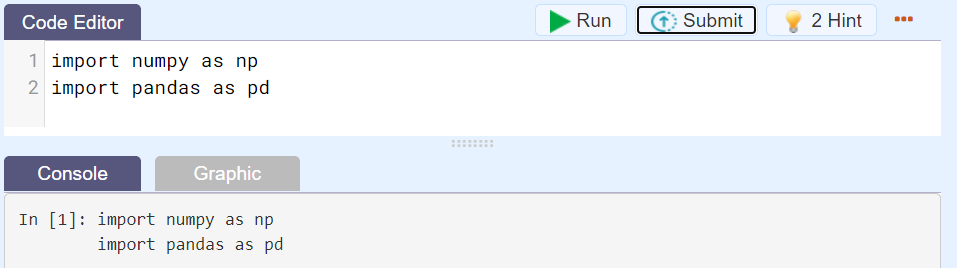
Sebelum dapat digunakan, library tersebut harus terlebih dahulu dipanggil ke dalam lingkungan Python. Command untuk memanggil library di Python menggunakan syntax (menggunakan huruf kecil):



**Alias** berfungsi sebagai pengganti nama library, sehingga menghemat komputasi saat function dari library tersebut dipanggil.

#### Tugas Praktek

Cobalah untuk mengimport library **numpy** dan **pandas** menggunakan alias **np** dan **pd** masing-masingnya.



## CHAPTER 2

### Membaca file dari Excel atau CSV sebagai data frame

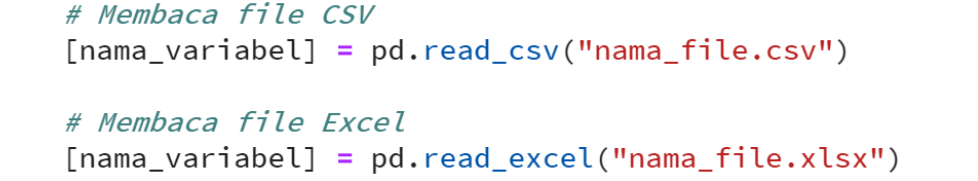
“Aksara, sudah cukup paham soal Library dalam Python?” sahut Andra tiba-tiba mengagetkanku yang sedang fokus mengerjakan kuis-kuis soal. Sudah kedua kalinya Andra muncul tiba-tiba seperti ini. Berbeda sekali dengan Senja yang biasanya penuh sapaan hangat di awal.

“Sudah, ini baru mau lanjut ke Exploratory Data Analysis. Ada apa?”

“Nah, pas sekali. Saya sebentar lagi mau pergi rapat dan sebelum itu saya hanya mau memastikan proses pembelajaran kamu ke depan. Jangan lupa kerjakan soal praktiknya karena pasti akan saya review.”

Sudah kuduga, Andra tidak akan duduk di sebelahku dan mengajariku langkah per langkah. Aku harus terbiasa dengan sistem belajar ini. Tetap semangat! Aku mengangguk dan mulai mencerna isi modul bab *Exploratory Data Analysis.*

Salah satu fungsi Pandas yaitu melakukan load data dari CSV atau Excel file. Syntax yang digunakan untuk melakukan operasi tersebut, yaitu:

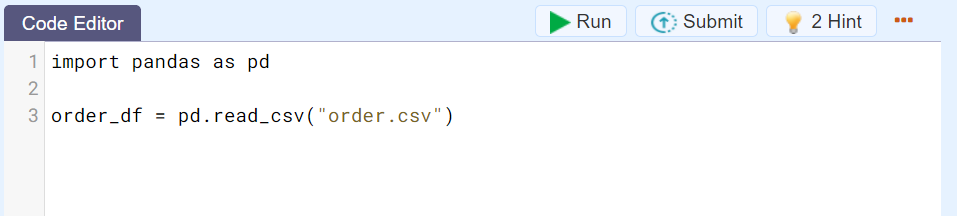


Nama variabel (**[nama\_variabel]**) dari contoh diatas menunjukkan nama variabel dari dataframe untuk menampung data dari datasets tersebut!

#### Tugas Praktek

Cobalah untuk mengimport dataset marketplace ABC dari **order.csv** dan disimpan ke dalam dataframe bernama **order\_df**.

Notes : untuk dataset diinput dari link berikut https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/order.csv



### Inspeksi struktur data frame

Setelah melakukan proses loading dataframe ke dalam Python. Hal selanjutnya sebelum memulai analisis tentunya mengerti struktur dataset tersebut. Sehingga langkah selanjutnya dari pre - analisis biasanya dilakukan untuk

1. melihat struktur data frame,
2. melihat preview data dari dataframe tersebut, dan
3. membuat summary data sederhana dari dataset.

### Melihat struktur kolom dan baris dari data frame

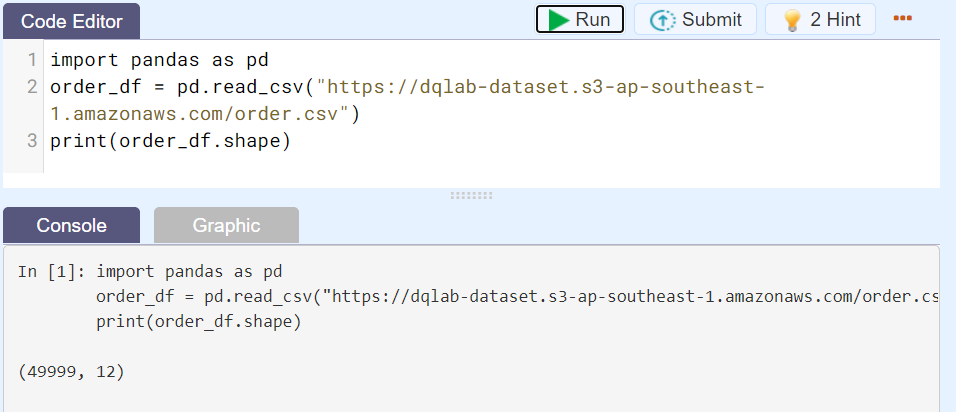
Hal pertama dalam mengerti struktur dari dataframe adalah informasi mengenai berapa size dari dataframe yang akan digunakan termasuk berapa jumlah kolom dan jumlah baris data frame tersebut.

Dalam kasus ini, aku dapat menggunakan fungsi .shape pada suatu dataframe. Syntaxnya dinyatakan dengan:



#### Tugas Praktek

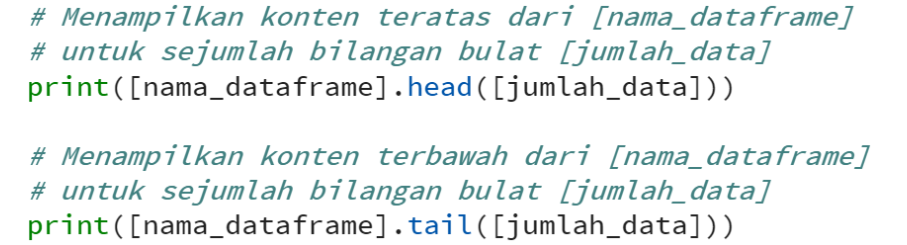
Cobalah untuk order dataframe dengan menuliskan syntax Python untuk melihat struktur dari **order\_df** dengan menggunakan fungsi **shape!**



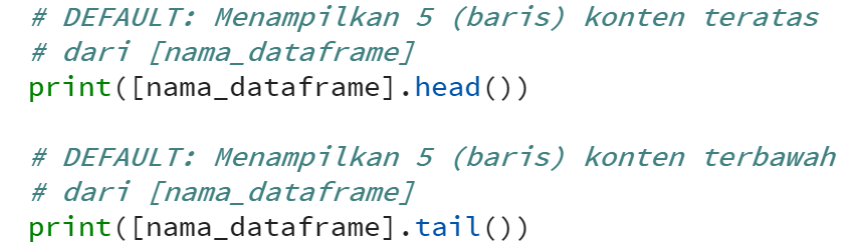
Di sini, terlihat bahwa dataframe order\_df memiliki 49999 baris serta 12 kolom.

### Melihat preview data dari data frame

Selanjutnya, untuk mendapatkan gambaran dari konten dataframe tersebut. Kita dapat menggunakan function **head** dan **tail,** dengan syntax:

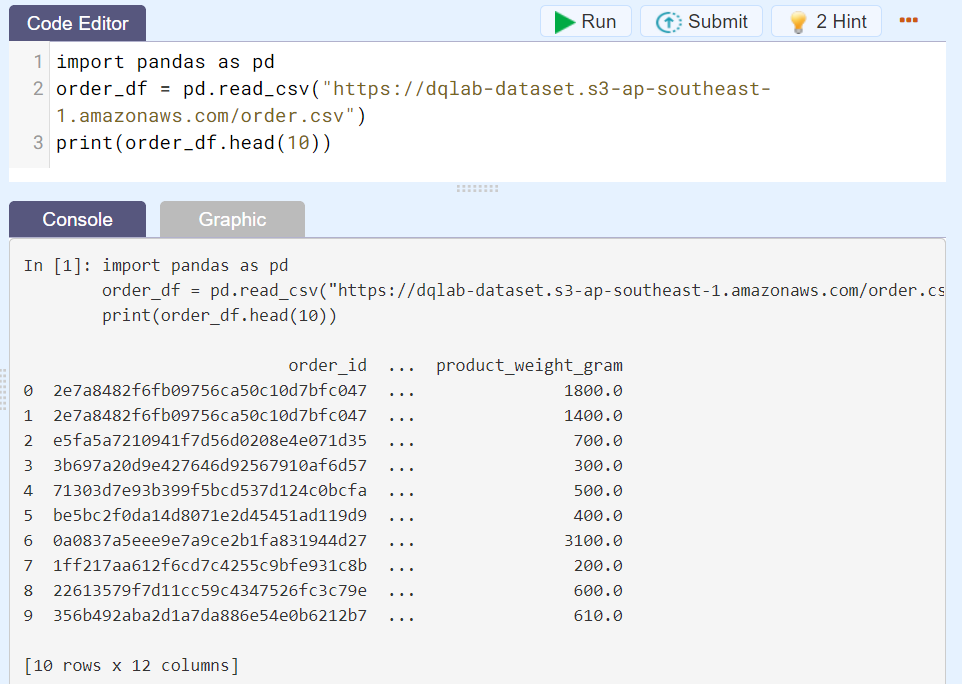


Jika [jumlah\_data] pada function **head** dan **tail** dikosongkan maka secara default akan di ditampilkan sebanyak 5 (lima) baris saja. Sehingga bisa ditulis sebagai berikut:



#### Tugas Praktek

Cobalah untuk check bagaimana contoh data dari dataframe tersebut dengan fungsi **head** dengan **limit 10 baris**!



### Statistik Deskriptif dari Data Frame - Part 1

Statistik deskriptif atau summary dalam Python - Pandas, dapat diperoleh dengan menggunakan fungsi **describe()**, yaitu:



Function describe dapat memberikan informasi mengenai nilai rataan, standar deviasi dan IQR (interquartile range).

Ketentuan umum:

* Secara umum function **describe()** akan secara otomatis mengabaikan kolom category dan hanya memberikan summary statistik untuk kolom berjenis numerik.
* Kita perlu menambahkan argument bernama **include = "all"** untuk mendapatkan summary statistik atau statistik deskriptif dari kolom numerik dan karakter.

Yaitu :



**Contoh penggunaan describe() di Pandas!**

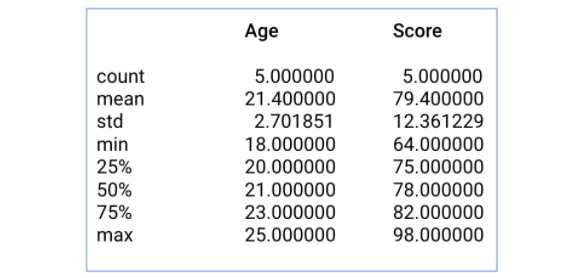
Terdapat dataframe Pandas dengan nama **nilai\_skor\_df** dengan informasi seperti gambar dibawah :



dengan menggunakan fungsi describe pada **nilai\_skor\_df**



Menghasilkan :

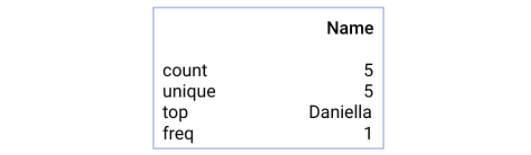


### Statistik Deskriptif dari Data Frame - Part 2

Jika ingin mendapatkan summary statistik dari kolom yang tidak bernilai angka, maka aku dapat menambahkan command **include=["object"]** pada syntax **describe()**.



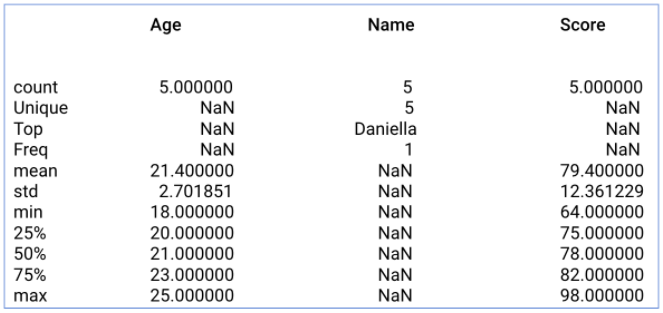
Hasil:



Function **describe()** dengan **include="all"** akan memberikan summary statistic dari semua kolom. Contoh penggunaannya:

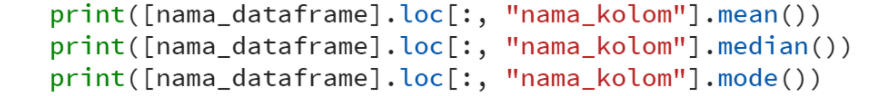


Hasil:

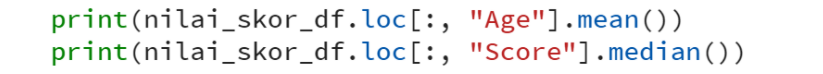


### Statistik Deskriptif dari Data Frame - Part 3

Selanjutnya, untuk mencari rataan dari suatu data dari dataframe. Aku dapat menggunakan syntax **mean, median,** dan **mode** dari Pandas.



***Contoh penggunaan:***



Memberikan hasil:

21.4 # Mean  
78 # Median

#### Tugas Praktek

Waktu sebentar lagi menunjukkan pukul dua belas siang. Sebentar lagi jam makan. Latihan praktik tadi sudah cukup jelas buatku, aku pun mulai membereskan modul dan hendak menutup laptop. Tapi kegiatanku terhenti ketika Andra sudah kembali dari rapatnya dan segera menghampiriku, “Aksara, kamu sempat mengevaluasi performa cabang A bukan?”

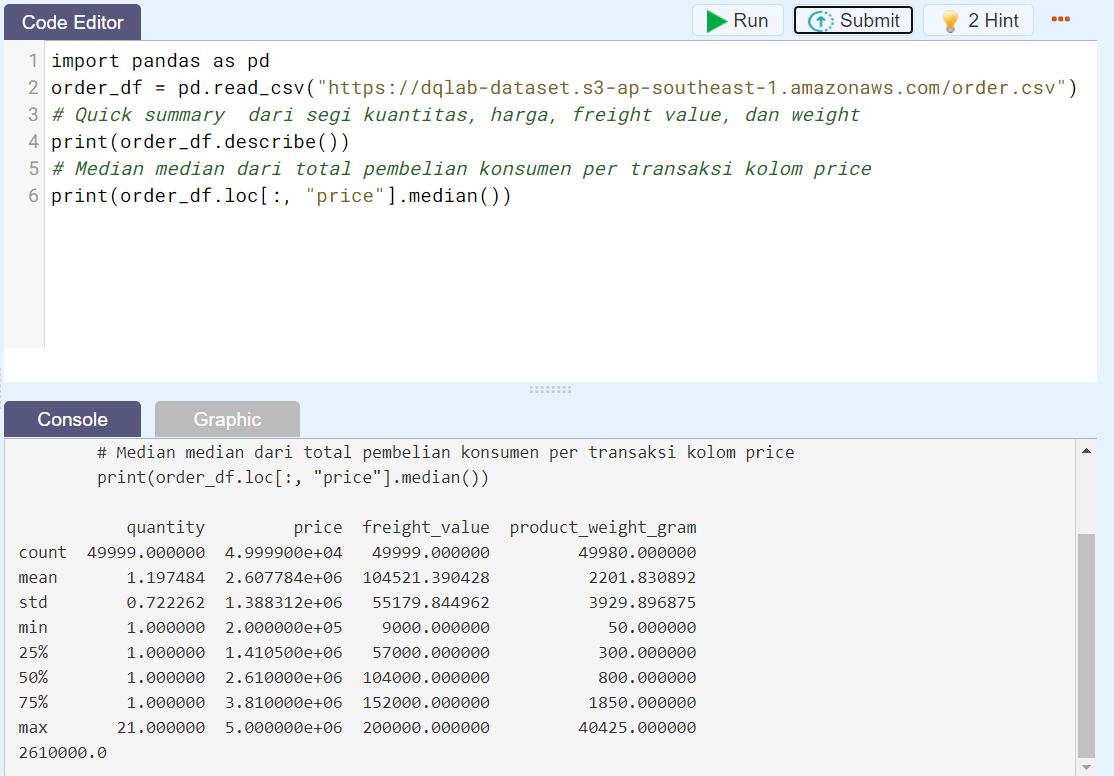
“Iya, terakhir aku sedang cek dari sisi dataframe **order\_df.** Sudah mau diminta hasilnya?” Aku melirik ke arah Andra yang sedang mencoret-coret buku catatannya sembari berdiri dari balik meja kerjaku.

“Bukan, ini ada tambahan quick summary dari segi kuantitas, harga, freight value, dan weight yang dibeli konsumen. Ada juga nih median dari total pembelian konsumen per transaksi. Jadi, bisa sekalian kamu analisis,” ujar Andra sembari merobek halaman buku catatannya dan memberikannya padaku.

“Tolong yah, Aksara. Soalnya sehabis ini saya ada rapat lanjutan lagi di luar kantor, mungkin baru balik sore nanti unutk kita bahas bersama hasilnya.”

“Siap,” jawabku yakin. Ini waktunya aku menunjukkan pada Andra kalau aku bisa!

Aku mengambil selotip dan menempelkan catatan dari Andra tadi di samping layar laptop sembari mulai membuat syntaks Phyton di code editor.



## CHAPTER 3

### Mengenal dan Membuat Distribusi Data dengan Histogram

Tugas tambahan dari Andra tadi sudah kuselesaikan, selagi menunggu Andra kembali. Aku coba lanjut untuk menyimak pembahasan lanjutan mengenai *Exploratory Data Analysis* terlebih dulu.

Histogram merupakan salah satu cara untuk mengidentifikasi sebaran distribusi dari data. Histogram adalah grafik yang berisi ringkasan dari sebaran (dispersi atau variasi) suatu data. Pada histogram, tidak ada jarak antar batang/bar dari grafik. Hal ini dikarenakan bahwa titik data kelas bisa muncul dimana saja di daerah cakupan grafik. Sedangkan ketinggian bar sesuai dengan frekuensi atau frekuensi relatif jumlah data di kelas. Semakin tinggi bar, semakin tinggi frekuensi data. Semakin rendah bar, semakin rendah frekuensi data.

***Syntax umum:***



Atau

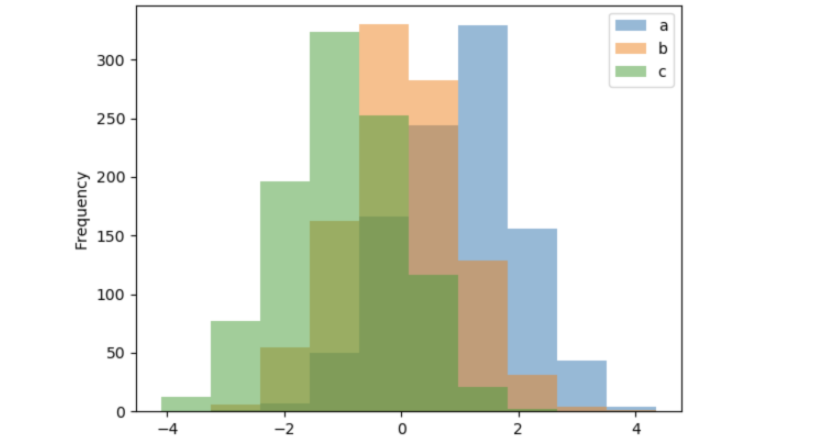


**\*CATATAN**

**Jika menggunakan xlabelsize & ylabelsize maka tidak bisa menggunakan syntax yang ada .plot. , maka langsung .hist agar tidak eror. Sebab syntax yang menggunakan .plot. histogramnya tidak ada garis pembatas antar bar. Lebih jelasnya bisa dicoba di command**

Beberapa atribut penting dalam histogram pandas:

* *bins* = jumlah\_bins dalam histogram yang akan digunakan. Jika tidak didefinisikan jumlah\_bins, maka function akan secara default menentukan jumlah\_bins sebanyak 10.
* *by* = Kolom di DataFrame untuk di group by. (valuenya berupa nama column di dataframe tersebut).
* *alpha* = Menentukan opacity dari plot di histogram. (value berupa range 0.0 - 1.0, dimana semakin kecil akan semakin kecil opacity nya)
* *figsize* = digunakan untuk menentukan ukuran dari plot histogram. Contoh: figsize=(10,12)



#### Tugas Praktek

Baru saja aku menyimak beberapa halaman isi modul, Andra sudah kembali ke kantor dari rapat lanjutannya. Tampaknya Andra sosok yang begitu sibuk di kantor. Ketika tahu langkahnya berbelok menuju tempatku, aku dengan sigap menyiapkan laptop untuk menunjukkan hasil kerjaku padanya.

“Pekerjaan tadi sudah selesai, Ndra. Bagaimana?”

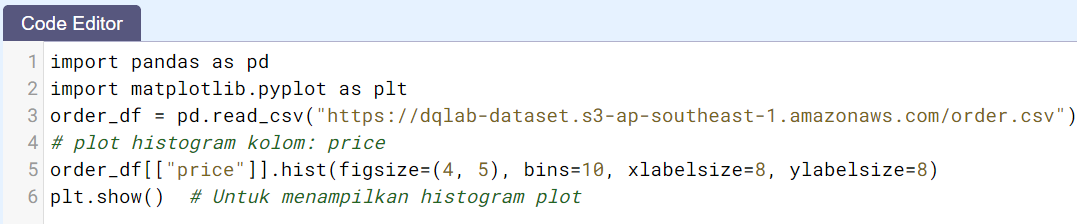
“Udah lebih rapi, ini nanti saya kasih ke kepala cabang untuk pertimbangan performa saat rapat,” ujar Andra cepat. Kulihat ia masih sibuk karena sesekali mengecek notifikasi ponselnya.

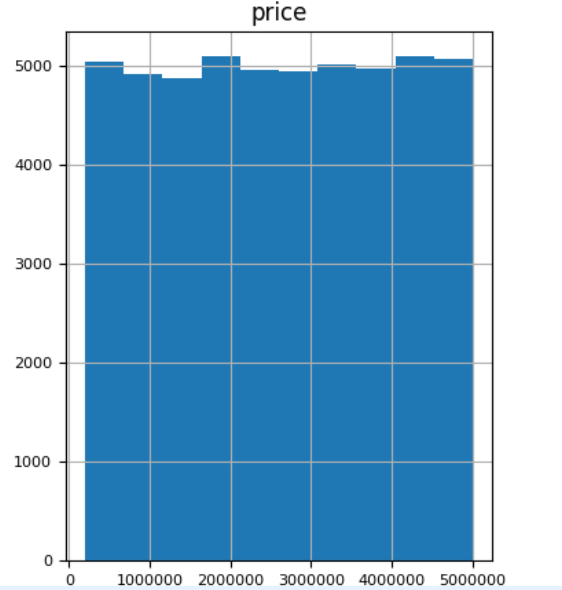
Baru saja aku mau menutup program dan pergi ke pantry, Andra kembali memanggilku, “Satu lagi, Aksara.”

“Ada data tambahan lagi?” tebakku.

“Kamu buatkan price distributionnya juga ya dari pembelian produk di cabang tadi. Saya rasa itu penting untuk diberitahu ke mereka. Bikin saja dalam bentuk histogram **price**,” saran Andra.

Aku pun kembali merebahkan diri di bangku dan mengutak-atik dataset **order\_df** dengan jumlah **bins=10**:

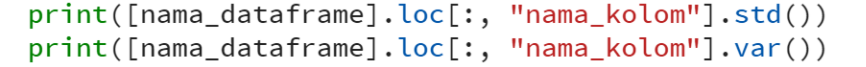




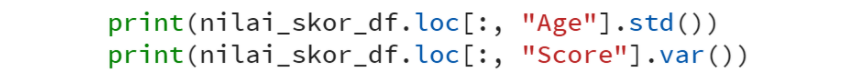
### Standar Deviasi dan Varians pada Pandas

Varians dan standar deviasi juga merupakan suatu ukuran dispersi atau variasi. Standar deviasi merupakan ukuran dispersi yang paling banyak dipakai. Hal ini mungkin karena standar deviasi mempunyai satuan ukuran yang sama dengan satuan ukuran data asalnya. Sedangkan varians memiliki satuan kuadrat dari data asalnya (misalnya cm^2).

Syntax dari standar deviasi dan varians pada Pandas:



Contoh penggunaan pada dataframe **nilai\_skor\_df:**



Hasil:

2.701851217  
152.8

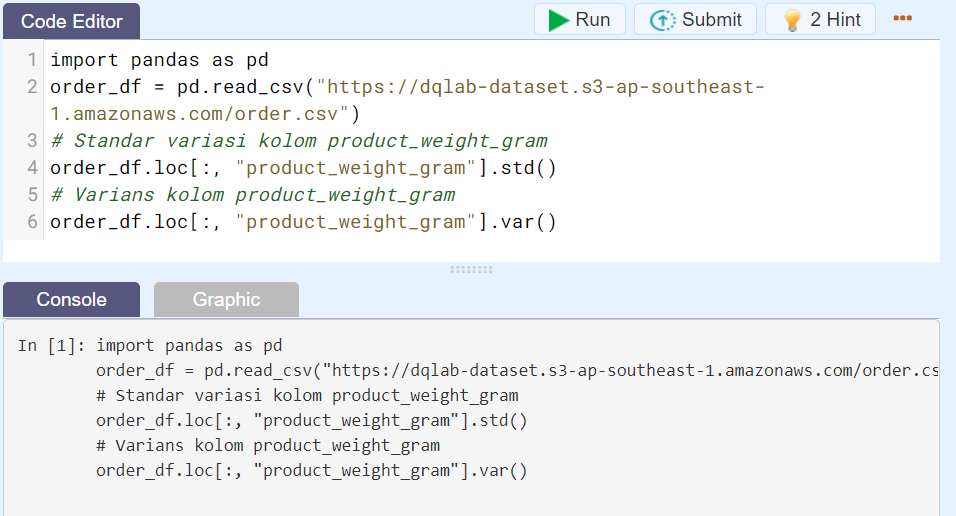
#### Tugas Praktek

“Aksara, dapat kabar terbaru dari kepala cabang. Tolong tampilkan data persebaran dari product\_weight\_gram penjualan cabang tadi ya,” ujar Andra kembali.

Wah, sehari bersama Andra adalah hari penuh pekerjaan beruntun. “Siap, Bos!” candaku.

“Enggak, hahaha. Sudah tahu bukan caranya bagaimana?” Baru pertama kali aku mendengar Andra tertawa! Kupikir ia tipe yang serius sekali.

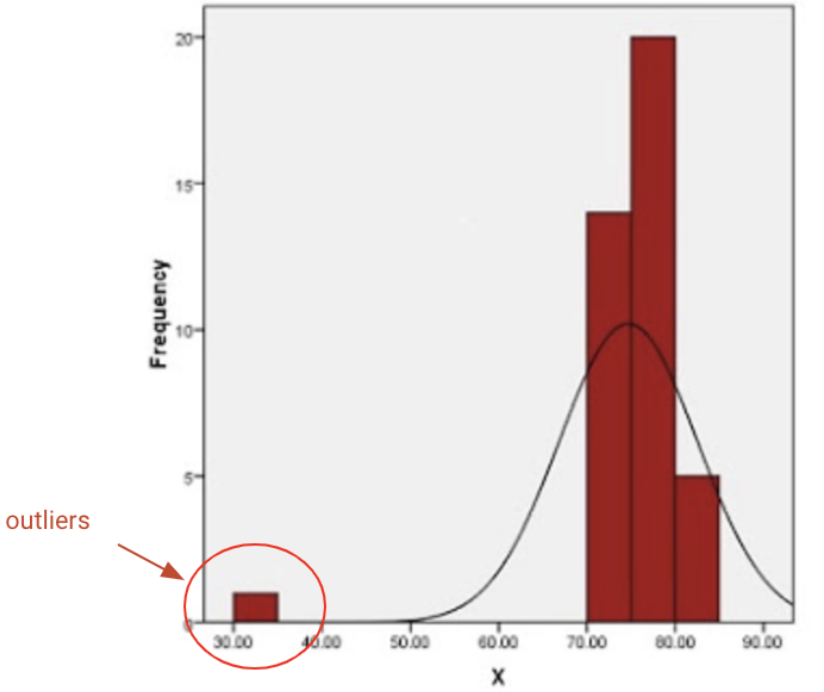
Aku pun menunjukkan layar laptopku yang sedang menggunakan standar deviasi dan variance untuk menganalisis lebar persebaran distribusi tersebut. Andra tersenyum puas melihat pilihanku yang ini:



### Menemukan Outliers Menggunakan Pandas

Sebelum menuju ke step by step dalam menemukan **outliers,** sedikit intermezo dahulu mengenai definisi dari **outliers.**

**Outliers** merupakan data observasi yang muncul dengan nilai-nilai ekstrim. Yang dimaksud dengan nilai-nilai ekstrim dalam observasi adalah nilai yang jauh atau beda sama sekali dengan sebagian besar nilai lain dalam kelompoknya.

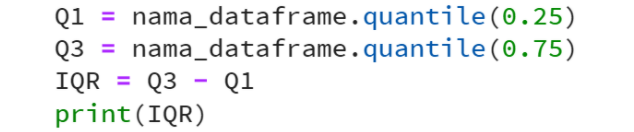


Pada umumnya, outliers dapat ditentukan dengan metric IQR (interquartile range).

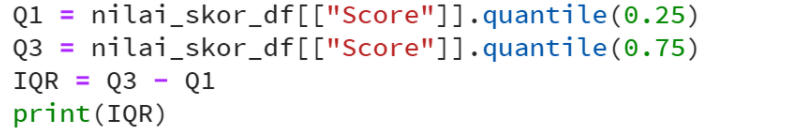
Rumus dasar dari IQR: Q3 - Q1. Dan data suatu observasi dapat dikatakan outliers jika memenuhi kedua syarat dibawah ini:

* data < Q1 - 1.5 \* IQR
* data > Q3 + 1.5 \* IQR

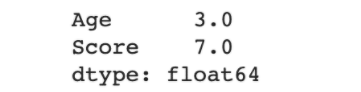
Syntax di Python:



Contoh case: mengidentifikasi IQR dari dataframe **nilai\_skor\_df**



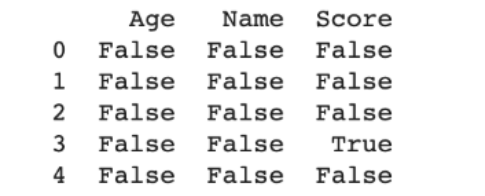
**Hasil:**



Karena saat ini memiliki skor IQR, saatnya untuk menentukan Outliers. Kode di bawah ini akan memberikan output dengan beberapa nilai True atau False. Titik data di mana terdapat False yang berarti nilai-nilai ini valid sedangkan True menunjukkan adanya Outliers.



Menghasilkan :

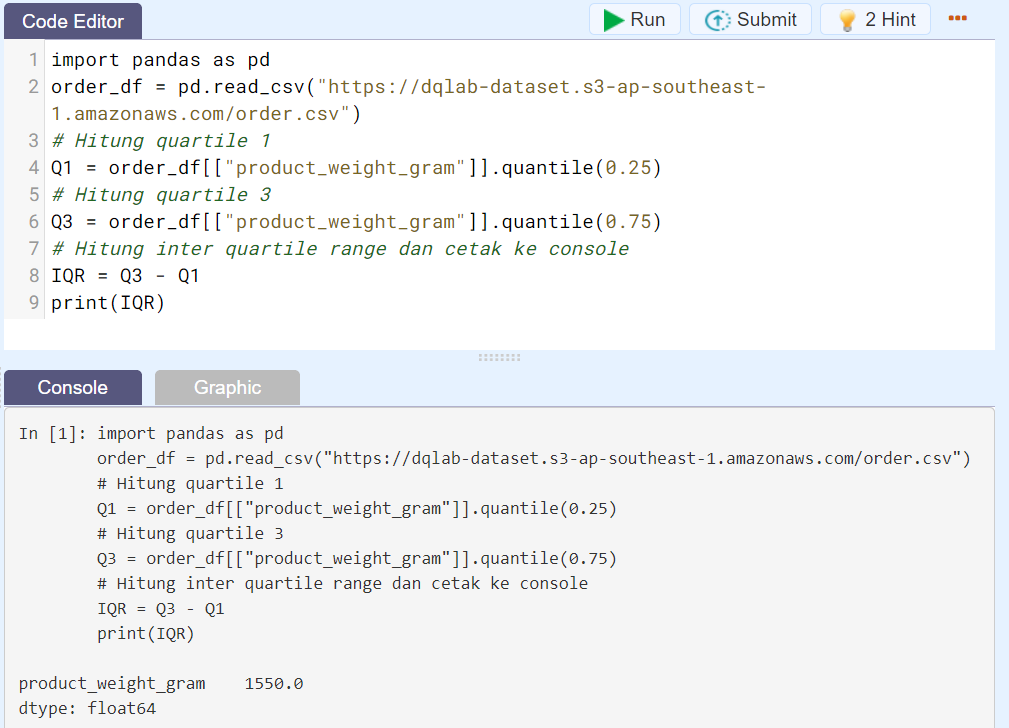


#### Tugas Praktek

Setelah mendapatkan persebaran dari dataset **order\_df** dan siap memberikannya pada Andra, aku terhenti sesaat.

Sepertinya aku perlu  menemukan batas IQR agar bisa menentukan outliers bagi kolom **product\_weight\_gram***.*

Kalau begitu, hasilnya jadi lebih lengkap. Aku pun kembali mengecek dan mengerjakan kode:



### Rename Kolom Data Frame

ada bagian ini, aku belajar cara mengganti nama kolom dataframe menggunakan Pandas. Mengganti nama kolom pada Pandas dapat dilakukan dengan 2 cara:

1. Menggunakan nama kolom.
2. Menggunakan indeks kolom.

**1. Rename menggunakan nama kolom**

Syntax:



Contoh penggunaan:



**2. Rename menggunakan indeks kolom**

Syntax:

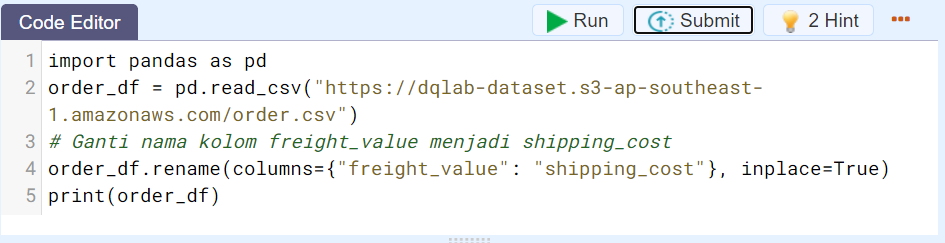
****

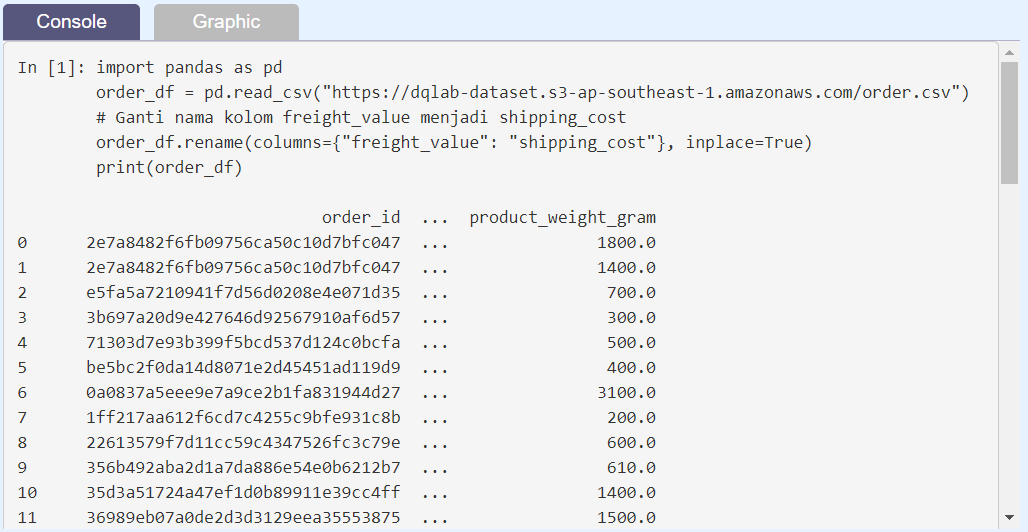
Contoh penggunaan:

****

#### Tugas Praktek

Cobalah untuk mengubah kolom **freight\_value** menjadi **shipping\_cost** dalam data frame **order\_df**, dengan menggunakan fungsi **rename()**.



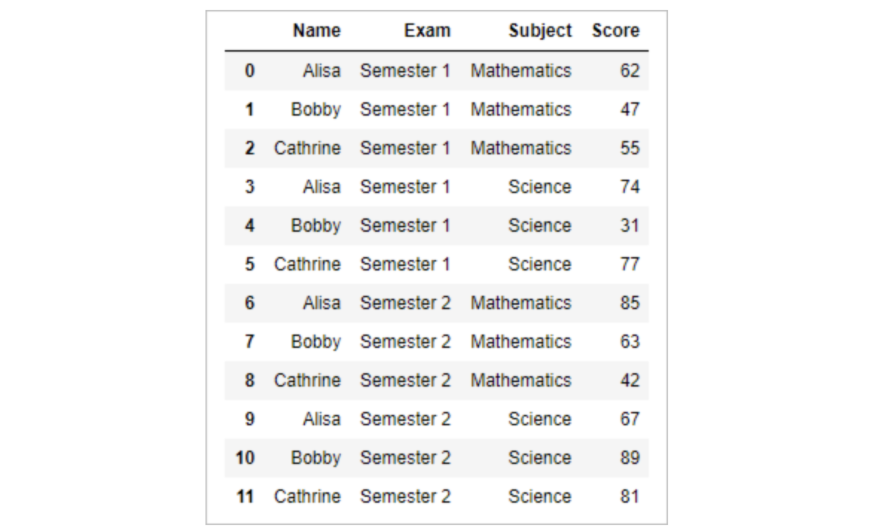


### .groupby menggunakan Pandas

Kegunaan .**groupby** adalah mencari summary dari data frame dengan menggunakan **aggregate** dari kolom tertentu.

***Contoh penggunaan:***

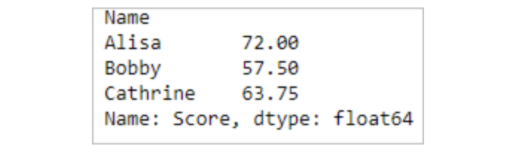
Diberikan dataset bernama **df** seperti pada gambar dibawah!



Penggunaan groupby:



Hasil:

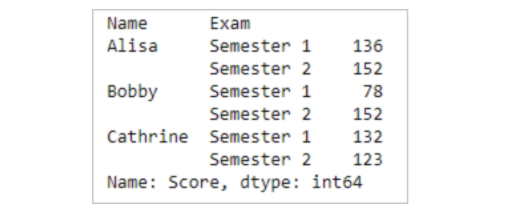


Penjelasan: komputasi diatas menggunakan kolom *‘Name’* sebagai *aggregate* dan kemudian menggunakan menghitung mean dari kolom *‘Score’* pada tiap - tiap aggregate tersebut.

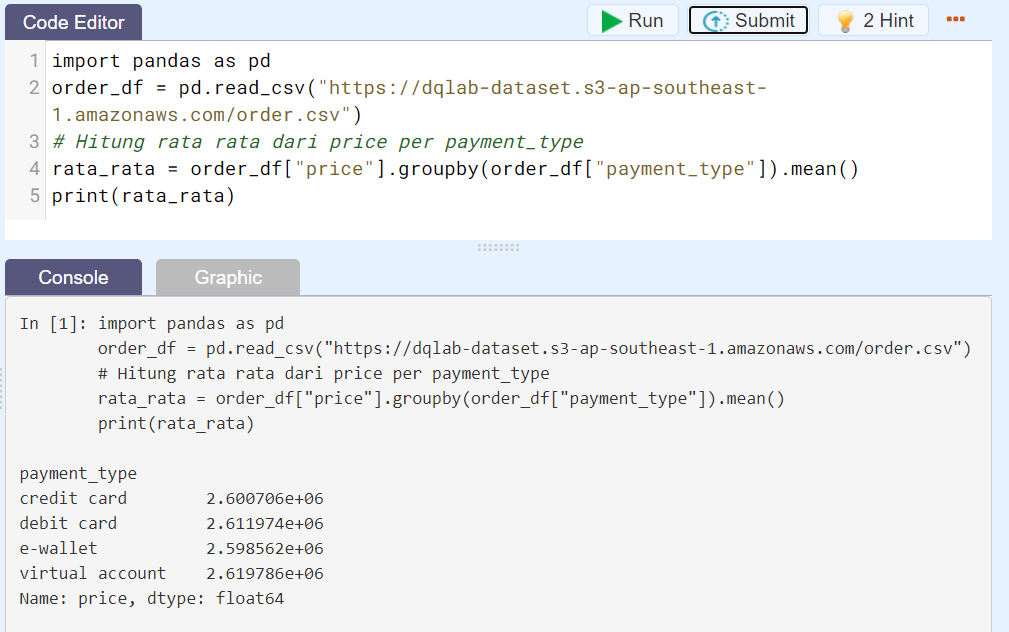
Contoh lainnya:



Hasil:



Penjelasan: komputasi diatas menggunakan kolom *‘Name’* dan *‘Exam’* sebagai *aggregate* dan kemudian menggunakan menghitung sum dari kolom *‘Score’* pada tiap - tiap aggregate tersebut.



### Sorting Menggunakan Pandas

**Sorting** adalah sebuah metode mengurutkan data berdasarkan syarat kolom tertentu, dan biasanya digunakan untuk melihat nilai maksimum dan minimum dari dataset. Library Pandas sendiri menyediakan fungsi sorting sebagai fundamental dari exploratory data analysis.

Syntax untuk operasi sorting pada Pandas:



**Contoh:**

Sorting terhadap dataset nilai\_skor\_df berdasarkan age!



Menghasilkan



Function tersebut akan secara default mengurutkan secara ascending (dimulai dari nilai terkecil), untuk dapat mengurutkan secara descending (nilai terbesar lebih dahulu), dapat menggunakan properti tambahan:



**Contoh:**

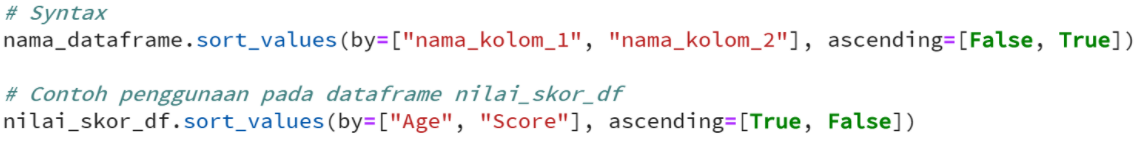
Sorting terhadap dataset nilai\_skor\_df berdasarkan age dimulai dari umur tertua!



Menghasilkan



Fungsi sorting di Pandas juga dapat dilakukan menggunakan lebih dari satu kolom sebagai syarat. Contoh nya pada skenario dibawah, akan mencoba mengaplikasikan fungsi Sorting menggunakan Age dan Score sekaligus:



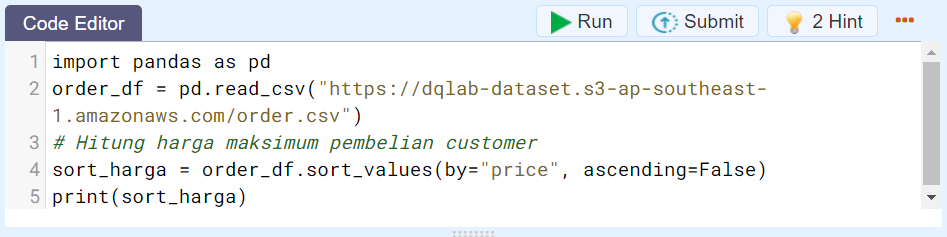
#### Tugas Praktek

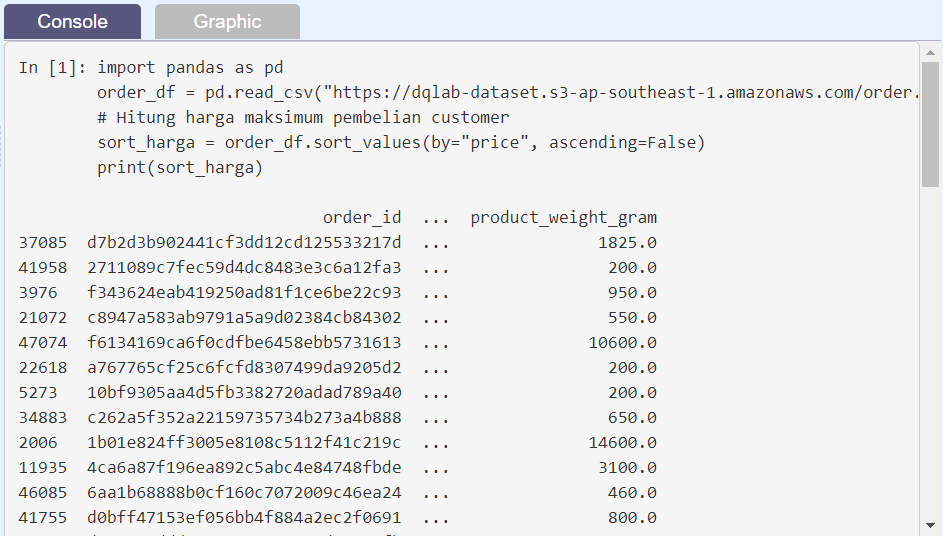
Tepat ketika kukira sudah selesai, ternyata masih ada tambahan! Lagi-lagi dari Andra. Aku tampaknya harus lebih terbiasa dengan cara bekerja seperti ini.

“Aksara, ini sudah menyeluruh dan lengkap. Tambahin satu lagi saja, tolong cari berapa harga maksimum pembelian customer di dataset **order\_df.**”

“Oke.”

Aku mulai mengerjakan rikues dari Andra dengan cara ini:





## CHAPTER 4

### Pendahuluan

Aku sedang menunggu review pekerjaan kemarin sembari membaca artikel-artikel menarik di media online ketika notifikasi email muncul. Dari Andra, dengan kategori pesan “urgent”.

### Tugas dari Andra

Aksara, bisa tolong bantu mengurus beberapa data penjualan dari dataset oder.csv? Saya sedang rapat dan bahan ini ditunggu dalam pembahasan cabang supermarket kita. Berikut ya detailnya:

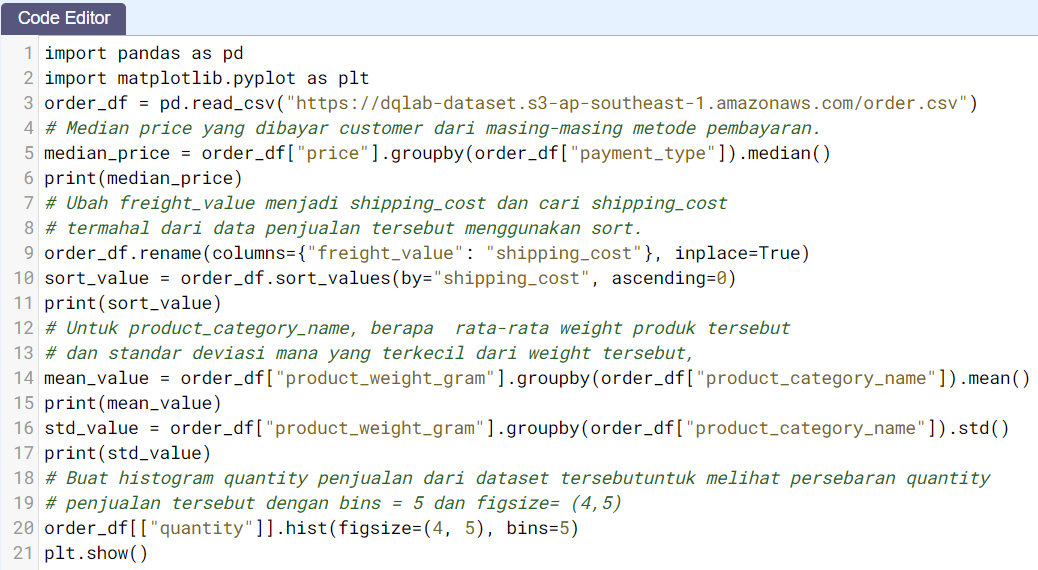
1. Median price yang dibayar customer dari masing-masing metode pembayaran.
2. Tentukan metode pembayaran yang memiliki basket size (rataan median price) terbesar.
3. Ubah freight\_value menjadi shipping\_cost dan cari shipping\_cost termahal dari data penjualan tersebut menggunakan sort.
4. Untuk product\_category\_name, berapa rata-rata weight produk tersebut dan standar deviasi mana yang terkecil dari weight tersebut,
5. Buat histogram quantity penjualan dari dataset tersebutuntuk melihat persebaran quantity penjualan tersebut dengan bins = 5 dan figsize= (4,5)

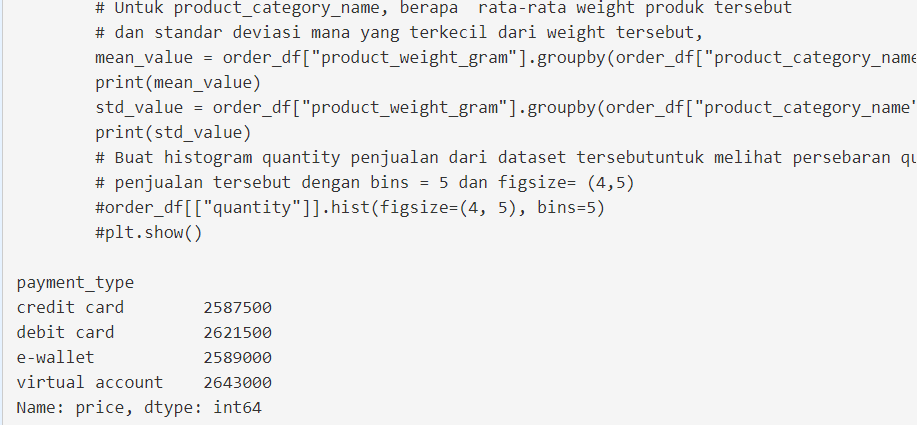
Khusus poin 4, tolong diperhatikan lebih ya, Aksara karena hasil analisisnya akan digunakan kepala cabang dalam menyusun strategi free ongkir.

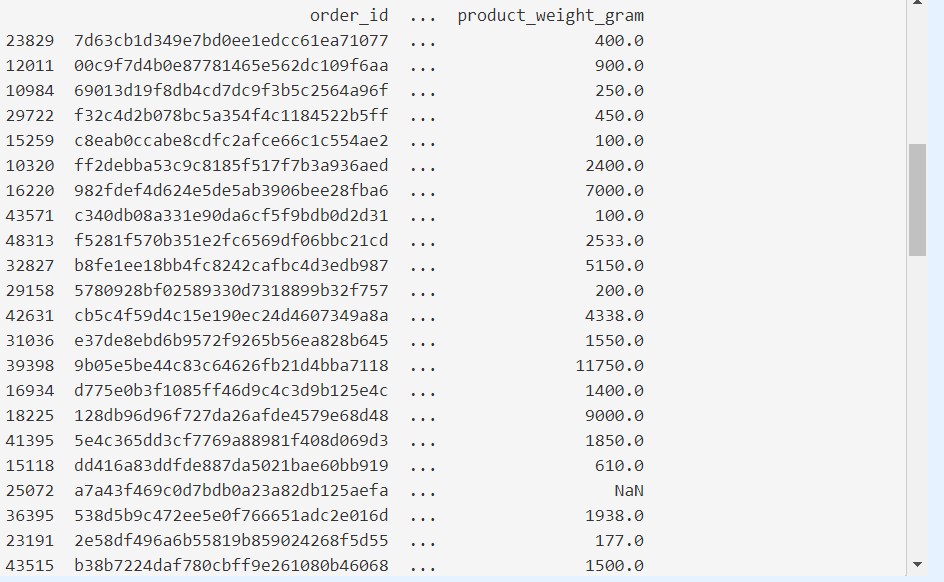
Kubalas email itu segera, *OK! Hasilnya akan selesai sebelum makan siang ya. You can count on me, hehehe.*

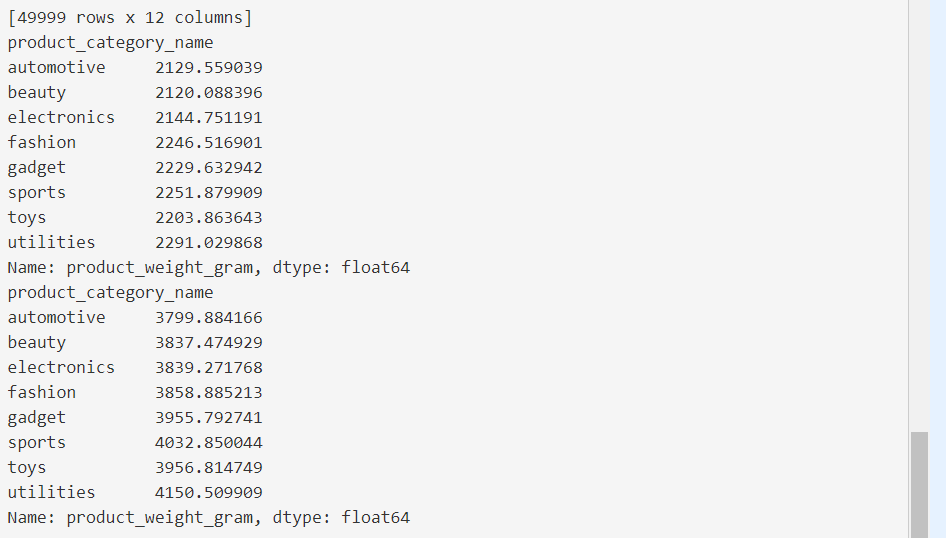
*Perhatian*: Semua string dinyatakan dalam kutipan "...".

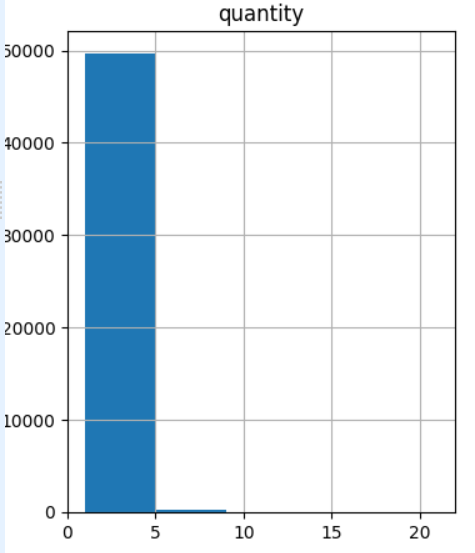
Hasil :











### Hasil Belajarku :)

Walau harus lembur, aku cukup puas dengan hasil kerjaku hari ini. Aku memandangi kode final yang sudah selesai kukerjakan. Diam-diam ada perasaan bangga menyelip di benakku! YES!

**Tidak terasa,**aku telah menyelesaikan modul ***Exploratory Data Analysis with Python for Beginner***. Dari materi-materi yang telah kupelajari dan praktekkan dalam modul ini aku telah mendapatkan pengetahuan (knowledge) dan praktek (skill) untuk:

* Memahami library Pandas, NumPy, dan Matplotlib dari Python.
* Mengetahui jenis - jenis tipe data dalam data frame Python
* Mampu membaca datasets dari Excel dan CSV
* Membuat summary data sederhana, mencakup distribusi, varians, dan mendeteksi outliers dari dataset.
* Latihan dalam membuat laporan bisnis sederhana menggunakan Python

**Semangat!**