

Challenge 1

Nama Lengkap : Yeni Rosandi

No. Registrasi : 01613821009

Kelas : B | Data Science

Nutrition Fact for McDonald's Menu

A. Studi Kasus

“Ray Kroc wanted to build a restaurant system that would be famous for providing food of consistently high quality and uniform methods of preparation. He wanted to serve burgers, buns, fries and beverages that tasted just the same in Alaska as they did in Alabama. To achieve this, he chose a unique path: persuading both franchisees and suppliers to buy into his vision, working not for McDonald's but for themselves, together with McDonald's. Many of McDonald's most famous menu items – like the Big Mac, Filet-O-Fish, and Egg McMuffin – were created by franchisees.”

Dalam kasus ini Ray Kroc sebagai stakeholder memiliki pertanyaan sebagai berikut:

1. How many calories does the average McDonald's value meal contain?
2. How much do beverages, like soda or coffee, contribute to the overall caloric intake?
3. Does ordered grilled chicken instead of crispy increase a sandwich's nutritional value?
4. What about ordering egg whites instead of whole eggs?
5. What is the least number of items could you order from the menu to meet one day's nutritional requirements?

B. Dataset

Dataset didapatkan dari McDonald's dengan judul “Nutrition Facts for McDonald's Menu” yang diperoleh melalui situs web Kaggle. Data tersebut memberikan data mengenai analisis nutrisi dari setiap items menu di McDona's Amerika Serikat, termasuk menu

breakfast, beef burgers, chicken and fish sandwiches, fries, salads, soda, coffee and tea, milkshakes, dan desserts. Dataset ini terdiri dari 24 kolom dan 261 baris, yang berarti data tersebut memiliki item sebanyak 260. Kategori yang paling banyak sebesar 37% adalah *Coffe & Tea*, kemudian kedua adalah Breakfast dengan persentase sebesar 16%, kemudian kategori lainnya sebesar 47%.

Sumber Data: <https://www.kaggle.com/mcdonalds/nutrition-facts>

C. Import Data

Kode program yang digunakan adalah dalam bentuk python, dan saya menggunakan Python Notebook pada platform IBM Cloud. Pertama-tama hal yang dilakukan adalah import library yang dibutuhkan beserta data csv yang ingin digunakan, kode program dan output dapat dilihat pada gambar dibawah ini:

```
In [1]: import types
import pandas as pd
from boto3.core.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_fac5a051a4cf46739b08369e9506134d = ibm_boto3.client(service_name='s3',
ibm_api_key_id='LjFZwumLPwdP_vPEEUz_U4Jt7kh0Y64-Fu5a52NSAJBi',
ibm_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
config=Config(signature_version='oauth'),
endpoint_url='https://s3-api.us-geo.objectstorage.service.networklayer.com')

body = client_fac5a051a4cf46739b08369e9506134d.get_object(Bucket='pythonnotebook-donotdelete-pr-xi6r2kerhpuc6z',Key='menu.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType(__iter__, body)

data = pd.read_csv(body)
data.head()
```

Out[1]:

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	...	Carbohydrates	Carbohydrates (% Daily Value)	Dietary Fiber	Dietary Fiber (% Daily Value)	Sugars	Protein
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	20	5.0	25	0.0	...	31	10	4	17	3	
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8.0	12	3.0	15	0.0	...	30	10	4	17	3	
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	35	8.0	42	0.0	...	29	10	4	17	2	

D. Cek Kualitas Data

Dibagian ini kualitas data dicek, dicari apakah ada yang null/kosong pada kolom/fitur, apakah ada angka yang tidak masuk akal dengan menggunakan kode program beserta outputnya yang dapat ditunjukkan dibawah ini:

```
In [2]: print(data.isnull().any())
```

Category	False
Item	False
Serving Size	False
Calories	False
Calories from Fat	False
Total Fat	False
Total Fat (% Daily Value)	False
Saturated Fat	False
Saturated Fat (% Daily Value)	False
Trans Fat	False
Cholesterol	False
Cholesterol (% Daily Value)	False
Sodium	False
Sodium (% Daily Value)	False
Carbohydrates	False
Carbohydrates (% Daily Value)	False
Dietary Fiber	False
Dietary Fiber (% Daily Value)	False
Sugars	False
Protein	False
Vitamin A (% Daily Value)	False
Vitamin C (% Daily Value)	False
Calcium (% Daily Value)	False
Iron (% Daily Value)	False
dtype:	bool

Dapat dilihat nilai yang dikembalikan adalah *false*, sehingga dapat disimpulkan tidak ada nilai *null* pada dataset tersebut.

E. Analisis Data

a) Analisis 1

How many calories does the average McDonald's value meal contain?

Untuk menjawab pertanyaan tersebut, kita dapat menggunakan fungsi `describe()`. `Describe` berfungsi untuk melihat ringkasan statistik dataset tersebut. Sehingga kode program dan output yang dikeluarkan adalah:

```
ln [3]: print(data.describe())
```

	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	\
count	260.000000	260.000000	260.000000	260.000000	
mean	368.269231	127.096154	14.165385	21.815385	
std	240.269886	127.875914	14.205998	21.885199	
min	0.000000	0.000000	0.000000	0.000000	
25%	210.000000	20.000000	2.375000	3.750000	
50%	340.000000	100.000000	11.000000	17.000000	
75%	500.000000	200.000000	22.250000	35.000000	
max	1880.000000	1060.000000	118.000000	182.000000	

	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	Cholesterol	\
count	260.000000	260.000000	260.000000	260.000000	
mean	6.007692	29.965385	0.203846	54.942308	
std	5.321873	26.639209	0.429133	87.269257	
min	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	4.750000	0.000000	5.000000	
50%	5.000000	24.000000	0.000000	35.000000	
75%	10.000000	48.000000	0.000000	65.000000	
max	20.000000	102.000000	2.500000	575.000000	

	Cholesterol (% Daily Value)	Sodium	...	Carbohydrates	\
count	260.000000	260.000000	...	260.000000	
mean	18.392308	495.750000	...	47.346154	
std	29.091653	577.026323	...	28.252232	
min	0.000000	0.000000	...	0.000000	
25%	2.000000	107.500000	...	30.000000	
50%	11.000000	190.000000	...	44.000000	
75%	21.250000	865.000000	...	60.000000	
max	192.000000	3600.000000	...	141.000000	

	Carbohydrates (% Daily Value)	Dietary Fiber	\
count	260.000000	260.000000	
mean	15.780769	1.630769	
std	9.419544	1.567717	
min	0.000000	0.000000	
25%	10.000000	0.000000	
50%	15.000000	1.000000	
75%	20.000000	3.000000	
max	47.000000	7.000000	

	Dietary Fiber (% Daily Value)	Sugars	Protein	\
count	260.000000	260.000000	260.000000	
mean	6.530769	29.423077	13.338462	
std	6.307057	28.679797	11.426146	
min	0.000000	0.000000	0.000000	
25%	0.000000	5.750000	4.000000	
50%	5.000000	17.500000	12.000000	
75%	10.000000	48.000000	19.000000	
max	28.000000	128.000000	87.000000	

	Vitamin A (% Daily Value)	Vitamin C (% Daily Value)	\
count	260.000000	260.000000	
mean	13.426923	8.534615	
std	24.366381	26.345542	
min	0.000000	0.000000	
25%	2.000000	0.000000	
50%	8.000000	0.000000	
75%	15.000000	4.000000	
max	170.000000	240.000000	

	Calcium (% Daily Value)	Iron (% Daily Value)	
count	260.000000	260.000000	
mean	20.973077	7.734615	
std	17.019953	8.723263	
min	0.000000	0.000000	
25%	6.000000	0.000000	
50%	20.000000	4.000000	
75%	30.000000	15.000000	
max	70.000000	40.000000	

Berdasarkan data yang didapatkan, terdapat data yang harus diperhatikan untuk menjawab pertanyaan tersebut, yaitu pada bagian:

	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	\
count	260.000000	260.000000	260.000000	260.000000	
mean	368.269231	127.096154	14.165385	21.815385	
std	240.269886	127.875914	14.205998	21.885199	
min	0.000000	0.000000	0.000000	0.000000	
25%	210.000000	20.000000	2.375000	3.750000	
50%	340.000000	100.000000	11.000000	17.000000	
75%	500.000000	200.000000	22.250000	35.000000	
max	1880.000000	1060.000000	118.000000	182.000000	

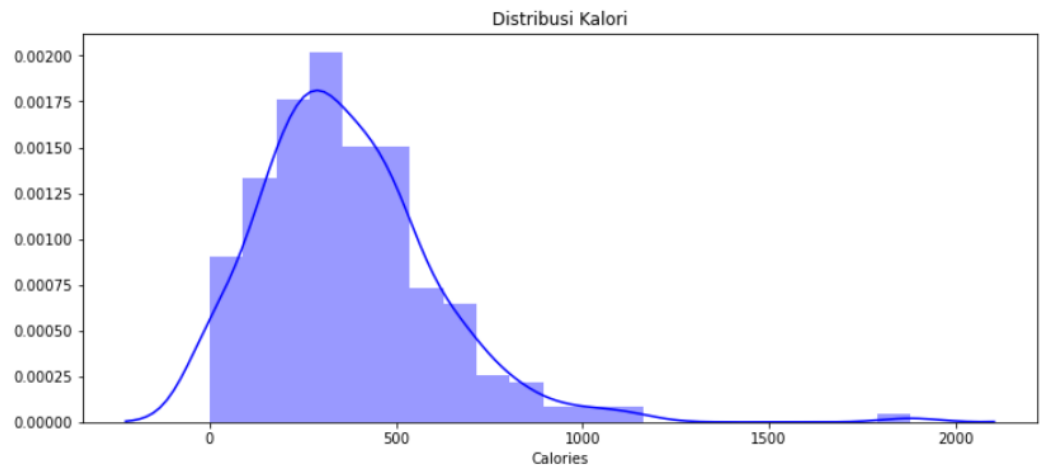
Sehingga berapa banyak **rata-rata kalori** yang terkandung dalam menu McDonald adalah sebanyak **368.269231 kalori**, dan kalori dari lemak sebanyak 127.096154 dari lemak total sebesar 14.165385. Distribusi kalori dapat digambarkan pada grafik berikut ini:

```
In [4]: import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12,5))
plt.title("Distribusi Kalori")
ax = sns.distplot(data["Calories"], color = 'blue')

print("Mean= ", data.Calories.mean())
print("Median= ", data.Calories.median())
```

Mean= 368.2692307692308
Median= 340.0



Grafik diatas diplot menggunakan library Matplotlib. Library Seaborn ini sendiri adalah library yang dibangun diatas Matplotlib sebagai tipe plot tambahan, agar grafik terlihat lebih cantik.

b) Analisis 2

How much do beverages, like soda or coffee, contribute to the overall caloric intake?

Untuk menjawab pertanyaan tersebut, kita terlebih dahulu buat data frame baru khusus untuk menampilkan data minuman, dari 13 jenis minuman yang diekstrak dari nama yang sama. Dengan menampilkan hanya kolom Category, Calories, Calories from Fat, dan menambahkan satu kolom dengan nama “Total Kalori” yang berisi nilai sum dari Calories dan Calories from Fat. Lalu dataframe tersebut di grup menjadi 3 kategori dan dicari nilai rata-ratanya untuk mendapatkan nilai keseluruhan. Kode program beserta outputnya dapat dilihat pada gambar berikut ini:

```

In [114]: #Filter Kategori Minuman
kalori_minuman = data[data['Category'].str.contains('Beverages|Coffee & Tea|Smoothies & Shakes')]

#Menambahkan kolom total kalori dari kolom kalori dan kalori lemak
kalori_minuman = kalori_minuman.iloc[:,0:5]
kalori_minuman['Jenis Minuman'] = kalori_minuman['Item'].str.extract("(Water|Juice|Milk Jug|Frappé|Tea|Coffee|Latte|M
kalori_minuman['Jenis Minuman'].fillna("Soda", inplace = True)
kalori_minuman['Total Kalori'] = kalori_minuman['Calories'] + kalori_minuman['Calories from Fat']

#Kategorikan minuman per Tema

mean_kalori_minuman = kalori_minuman.groupby('Jenis Minuman').mean().sort_values(by='Total Kalori')
mean_kalori_minuman

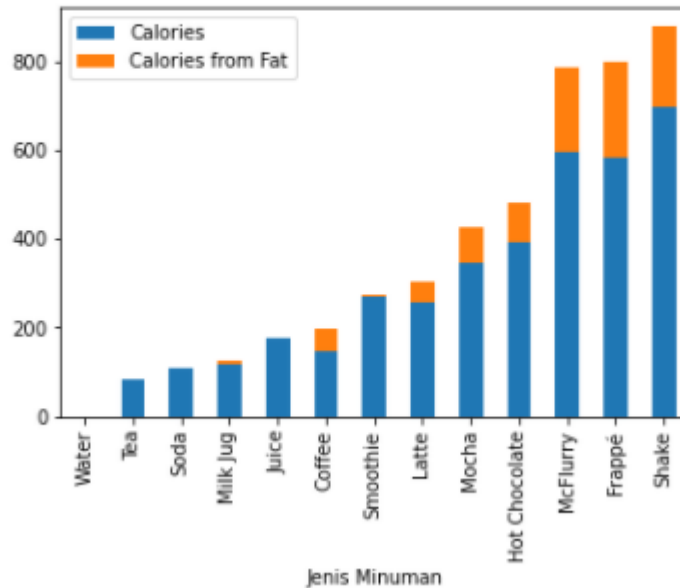
```

Out[114]:

	Calories	Calories from Fat	Total Kalori
Jenis Minuman			
Water	0.000000	0.000000	0.000000
Tea	82.500000	0.000000	82.500000
Soda	107.000000	0.000000	107.000000
Milk Jug	115.000000	10.000000	125.000000
Juice	175.000000	0.000000	175.000000
Coffee	145.555556	50.000000	195.555556
Smoothie	268.888889	7.222222	276.111111
Latte	257.333333	48.333333	305.666667
Mocha	347.916667	80.416667	428.333333
Hot Chocolate	393.333333	89.166667	482.500000
McFlurry	596.250000	191.250000	787.500000
Frappé	584.444444	215.555556	800.000000
Shake	699.090909	180.000000	879.090909

Output diatas adalah tabel rata-rata dalam bentuk tabel. Selanjutnya membuat visualisasi data agar data dapat lebih mudah dipahami. Visualisasi yang digunakan adalah barplot dengan kedua nilai dijadikan tumpang tindih yang dapat dilihat dibawah ini:

```
In [116]: mean_kalori_minuman.iloc[:,2].plot(kind='bar', stacked=True)
plt.title('Total Kalori pada Minuman')
plt.ylabel('Total Kalori')
plt.show()
```



Visualisasi data selesai. Sehingga kita dapat melihat bahwa setiap minuman memiliki kontribusi kalori sebesar:

- 1) Air adalah minuman yang tidak berkalori
- 2) Teh menduduki peringkat kedua dan yang pertama untuk minuman berkalori rendah dengan total kalori sebesar 82
- 3) Dilanjutkan dengan soda yang menduduki peringkat ke 3 dengan kalori sebesar 107
- 4) Peringkat ke-4 Milk Jug dengan total kalori sebesar 115
- 5) Peringkat ke-5 Jus dengan total kalori sebesar 175
- 6) Peringkat ke-6 Kopi dengan total kalori sebesar 195.556
- 7) Peringkat ke-7 Smoothie dengan total kalori sebesar 276.11
- 8) Peringkat ke-8 Latte dengan total kalori sebesar 305.667
- 9) Peringkat ke-9 Mocha dengan total kalori sebesar 428.333
- 10) Peringkat ke-10 Hot Chocolate dengan total kalori sebesar 482.500
- 11) Peringkat ke-11 McFlurry dengan total kalori sebesar 787.500

12) Peringkat ke-12 Frappe dengan total kalori sebesar 800.000

13) Dan total kalori paling tinggi pertama pada peringkat ke-13 adalah Shake dengan total kalori sebesar 979.091

c) Analisis 3

Does ordered grilled chicken instead of crispy increase a sandwich's nutritional value?

Untuk menjawab pertanyaan tersebut, terlebih dahulu buat data frame baru khusus untuk Sandwich, karena pada pertanyaan ini hanya membutuhkan data yang dikategorikan Sandwich (lebih spesifik). Kolom yang digunakan juga diseleksi sehingga kolom hanya terdiri dari nama item, tipe ayam, tipe daging, dan nutrisi yang terkandung di dalam items seperti lemak, kolesterol, sodium, karbohidrat, serat, vitamin A, vitamin C, kalsium, zat besi, dan total nilai harian. Sehingga dataframe yang dibuat adalah sebagai berikut:

```
In [69]: #Filter Kategori Sandwich
sandwich_menu = data[data['Item'].str.contains('Sandwich')]
#Filter Kolom
sandwich_menu = sandwich_menu.iloc[:, [1,6,8,11,13,15,17,20,21,22]]
#Buat kolom baru nilai total keseluruhan nilai nutrisi
sandwich_menu['Total Daily Value'] = sandwich_menu.sum(axis=1)
#Buat Kolom baru buat yg ekstrak Item tsb Crispy/Grilled
sandwich_menu['Chicken Type'] = sandwich_menu['Item'].str.extract("(Crispy|Grilled)")
#Ini untuk mengkategorikan jenis daging
sandwich_menu['Item Type'] = sandwich_menu['Item'].str.extract("(Classic|Club|Ranch|Bacon|Southern)")
sandwich_menu
```

Out[69]:

	Item	Total Fat (% Daily Value)	Saturated Fat (% Daily Value)	Cholesterol (% Daily Value)	Sodium (% Daily Value)	Carbohydrates (% Daily Value)	Dietary Fiber (% Daily Value)	Vitamin A (% Daily Value)	Vitamin C (% Daily Value)	Calcium (% Daily Value)	Total Daily Value	Chicken Type	Item Type
57	Premium Crispy Chicken Classic Sandwich	33	18	16	41	18	13	4	6	15	164	Crispy	Classic
58	Premium Grilled Chicken Classic Sandwich	13	9	22	34	14	13	4	8	15	132	Grilled	Classic
59	Premium Crispy Chicken Club Sandwich	51	44	29	59	19	14	8	20	30	274	Crispy	Club
60	Premium Grilled Chicken Club Sandwich	31	36	35	52	15	13	8	20	30	240	Grilled	Club
61	Premium Crispy Chicken Ranch BLT Sandwich	43	31	24	58	19	13	4	20	15	227	Crispy	Ranch

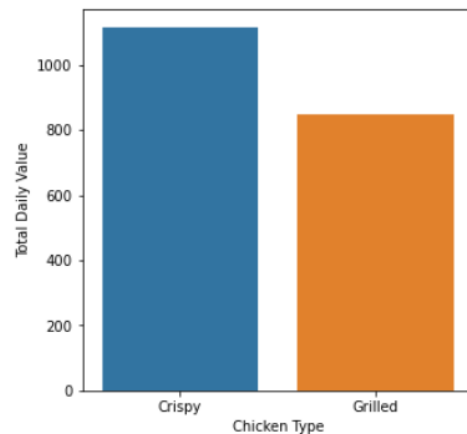
Setelah dataframe khusus sandwich dibuat maka dapat dimulai untuk plotting grafik untuk menganalisa atau mengetahui nilai total nutrisi harian per tipe ayam (Chicken Type).

```
In [81]: df_total= sandwich_menu.groupby('Chicken Type').sum()
df_total
```

Out[81]:

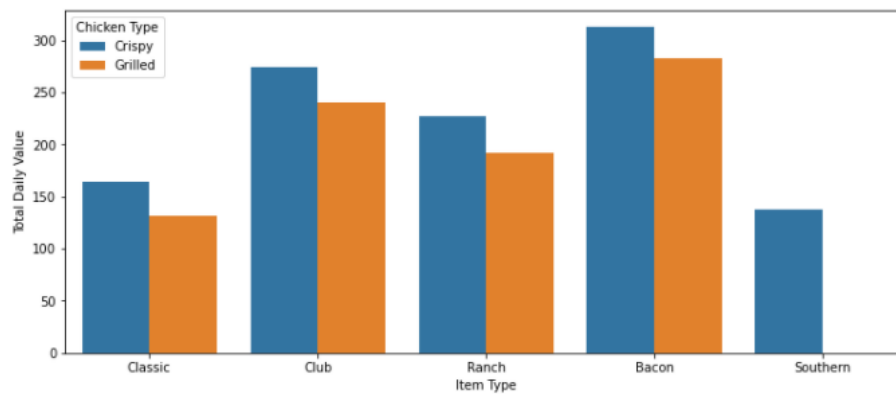
	Chicken Type	Total Fat (% Daily Value)	Saturated Fat (% Daily Value)	Cholesterol (% Daily Value)	Sodium (% Daily Value)	Carbohydrates (% Daily Value)	Dietary Fiber (% Daily Value)	Vitamin A (% Daily Value)	Vitamin C (% Daily Value)	Calcium (% Daily Value)	Total Daily Value
0	Crispy	215	159	114	268	92	62	28	73	105	1116
1	Grilled	106	109	124	202	60	54	24	78	90	847

```
In [82]: df_total.reset_index(inplace=True)
plt.figure(figsize=(5,5))
sns.barplot(df_total['Chicken Type'], df_total['Total Daily Value'], estimator=sum)
plt.show()
```



Data diatas terlihat Ayam Crispy lebih banyak nilai total nutrisi harian dibandingkan Ayam Bakar. Untuk melihat lebih spesifik lagi dapat dihat melalui grafik bar lagi namun lebih spesifik berdasarkan tipe item. Hasil yang didapatkan adalah:

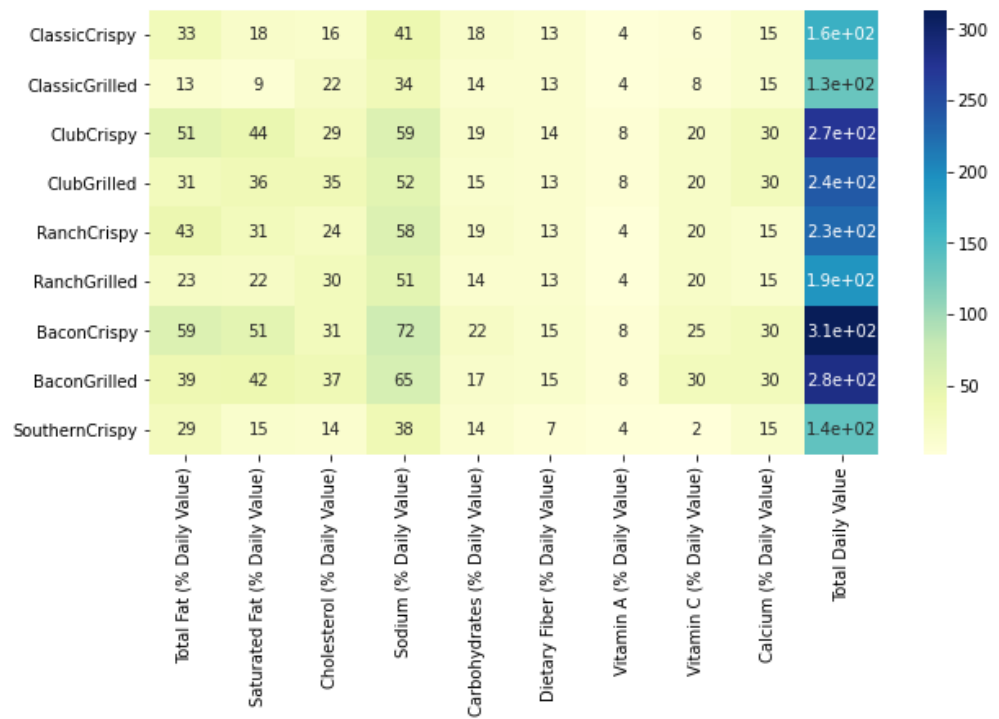
```
In [76]: plt.figure(figsize=(12,5))
sns.barplot(sandwich_menu['Item Type'], sandwich_menu['Total Daily Value'], hue=sandwich_menu['Chicken Type'])
plt.show()
```



Berdasarkan gambar diatas pada ke-5 tipe item terlihat jenis ayam Crispy lebih

banyak nutrisinya dibandingkan ayam bakar. Kemudian mari kita pecah lebih rinci lagi dengan menampilkan kandungan tiap nutrisi yang ada melalui heatmap plot. Hasil yang didapatkan adalah:

```
In [92]: #HeatMap
plt.figure(figsize=(10,5))
sns.heatmap(sandwich_menu.iloc[:,1:11], cmap='YlGnBu', annot=True,
            yticklabels=sandwich_menu['Item Type']+sandwich_menu['Chicken Type'])
plt.show()
```



Pada plot HeatMap ini dapat dilihat alasan mengapa Ayam Crispy memiliki total nilai harian yang lebih tinggi. Hal tersebut dikarenakan terlihat perbedaan yang sangat signifikan pada nilai Total Fat dimana antara Crispy dan Grilled memiliki selisih yang paling besar dibandingkan nutrisi lainnya yaitu sebesar 20%.

Sehingga pertanyaan “Does ordered grilled chicken instead of chicken crispy increase a sandwich's nutritional value?” adalah **Tidak**. Dikarenakan ayam bakar memiliki beberapa nilai nutrisi lebih rendah daripada ayam krispy.

d) Analisis 4

What about ordering egg whites instead of whole eggs?

Untuk menjawab pertanyaan tersebut, terlebih dahulu buat data frame baru khusus untuk menu yang memiliki kalimat “with egg white” atau “with egg”, pada kasus ini kita akan membandingkan nutrisi pada menu yang sama dengan tambahan telur yang berbeda (egg white atau egg). Pertama-tama mari kita lihat pada data csv tersebut ada berapa data yang memiliki kalimat yang diharapkan menggunakan fitur find:

No.	Menu with Egg/Egg White
1.	Bacon, Egg & Cheese Bagel with Egg Whites
2.	Bacon, Egg & Cheese Biscuit with Egg Whites (Large Biscuit)
3.	Bacon, Egg & Cheese Biscuit with Egg Whites (Regular Biscuit)
4.	Bacon, Egg & Cheese McGriddles with Egg Whites
5.	Big Breakfast with Egg Whites (Large Biscuit)
6.	Big Breakfast with Egg Whites (Regular Biscuit)
7.	Sausage Biscuit with Egg (Large Biscuit)
8.	Sausage Biscuit with Egg (Regular Biscuit)
9.	Sausage Biscuit with Egg Whites (Large Biscuit)
10.	Sausage Biscuit with Egg Whites (Regular Biscuit)
11.	Sausage McMuffin with Egg
12.	Sausage McMuffin with Egg Whites
13.	Sausage, Egg & Cheese McGriddles with Egg Whites

Data diatas menampilkan 13 data namun hanya 6 data yang memiliki menu yang dapat dibandingkan. Sehingga data frame yang dibuat adalah sebagai berikut:

```

In [177]: #Ambil data yang ada kata "with egg" dan 6 nama menu yg sama yg ingin dibandingkan di kolom itemnya
egg_menu = data[data['Item'].str.contains("with Egg")]
egg_menu = egg_menu[egg_menu['Item'].str.contains("Sausage McMuffin|Sausage Biscuit")]
#Select kolom & buat kolom baru berisi Total nilai nutrisi sehari-hari
egg_menu = egg_menu.iloc[:, [1,6,8,11,13,15,17,20,21,22,23]]
egg_menu['Total Daily Value'] = egg_menu.sum(axis=1)
#Menambah
egg_menu['Egg Type'] = egg_menu['Item'].str.extract("(Egg Whites)")
egg_menu['Egg Type'].fillna('Egg', axis=0, inplace=True)
egg_menu['Menu'] = egg_menu['Item'].str.extract("(Large Biscuit|Regular Biscuit|Sausage McMuffin)")

egg_menu=egg_menu.sort_values(by='Menu')
egg_menu

```

Out[177]:

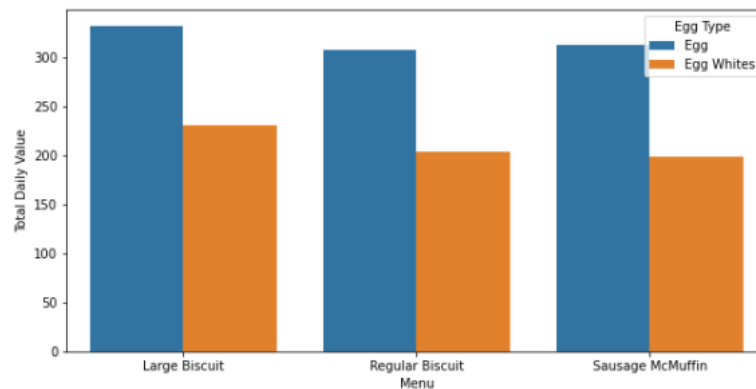
Item	Total Fat (% Daily Value)	Saturated Fat (% Daily Value)	Cholesterol (% Daily Value)	Sodium (% Daily Value)	Carbohydrates (% Daily Value)	Dietary Fiber (% Daily Value)	Vitamin A (% Daily Value)	Vitamin C (% Daily Value)	Calcium (% Daily Value)	Iron (% Daily Value)	Total Daily Value	Egg Type	Menu
Sausage Biscuit with Egg (Large Biscuit)	57	74	83	53	14	11	10	0	10	20	332	Egg	Large Biscuit
Sausage Biscuit with Egg Whites (Large Biscuit)	49	65	11	54	13	11	4	0	8	15	230	Egg Whites	Large Biscuit
Sausage Biscuit with Egg (Regular Biscuit)	50	71	83	49	12	6	6	0	10	20	307	Egg	Regular Biscuit
Sausage Biscuit with Egg Whites (Regular Biscuit)	42	62	11	49	11	6	0	0	8	15	204	Egg Whites	Regular Biscuit
Sausage McMuffin with Egg	43	52	95	36	10	17	15	0	30	15	313	Egg	Sausage McMuffin
Sausage McMuffin with Egg Whites	35	42	16	37	10	17	6	0	25	10	198	Egg Whites	Sausage McMuffin

Setelah dataframe dibuat maka tahap selanjutnya adalah visualisasi data. Kita akan membandingkan data menu dengan tipe egg dan egg white menggunakan barplot:

```

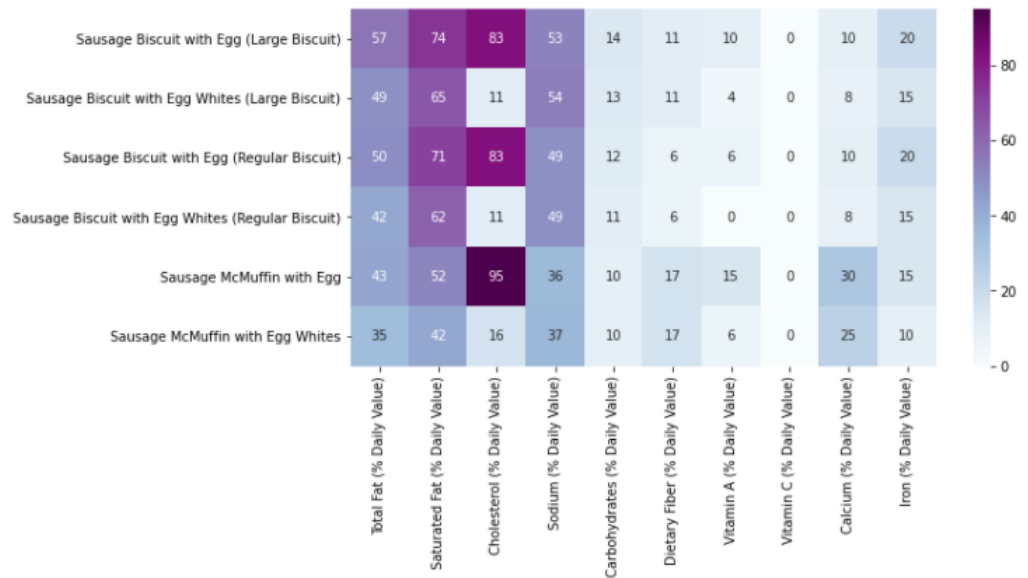
In [180]: plt.figure(figsize=(10,5))
sns.barplot(egg_menu['Menu'], egg_menu['Total Daily Value'], hue=egg_menu['Egg Type'])
plt.show()

```



Seperti grafik diatas dapat dilihat bahwa total nutrisi harian lebih banyak pada makanan yang diberikan telur utuh, bukan hanya telur putih. Untuk melihat alasan lebih rinci, kita dapat lihat pada HeatMap plot sebagai berikut:

```
In [181]: plt.figure(figsize=(10,5))
sns.heatmap(egg_menu.iloc[:,1:11], cmap='BuPu', annot=True,
            yticklabels=egg_menu['Item'])
plt.show()
```



Pada plot HeatMap ini dapat dilihat alasan mengapa Makanan dengan telur utuh memiliki total nilai nutrisi harian yang lebih tinggi. Hal tersebut dikarenakan terlihat perbedaan yang sangat signifikan pada nilai kolesterol dimana antara Egg dan Egg White memiliki selisih yang paling besar dibandingkan nutrisi lainnya yaitu sebesar 80%. Analisis ini membuktikan bahwa telur kuning yang ada pada telur mengandung kolesterol yang banyak.

Sehingga pertanyaan “*What about ordering egg whites instead of whole eggs?*” adalah **Tidak**. Dikarenakan telur putih memiliki nilai nutrisi yang lebih rendah dibandingkan telur yang utuh.

e) Analisis 5

What is the least number of items could you order from the menu to meet one day's nutritional requirements?

Untuk menjawab pertanyaan tersebut, terlebih dahulu buat data frame baru dengan kolom yang digunakan masih sama dengan yang sebelumnya dikarenakan pada analisis ini bertujuan untuk mencari kombinasi item yang memungkinkan untuk memenuhi kebutuhan nutrisi dalam sehari. Dataframe dan ringkasan statistik dapat dilihat sebagai berikut:

```
In [185]: least_order=data.iloc[:,[1,6,8,11,13,15,17,20,21,22,23]]
least_order.head(5)
```

Out[185]:

	Item	Total Fat (% Daily Value)	Saturated Fat (% Daily Value)	Cholesterol (% Daily Value)	Sodium (% Daily Value)	Carbohydrates (% Daily Value)	Dietary Fiber (% Daily Value)	Vitamin A (% Daily Value)	Vitamin C (% Daily Value)	Calcium (% Daily Value)	Iron (% Daily Value)
0	Egg McMuffin	20	25	87	31	10	17	10	0	25	15
1	Egg White Delight	12	15	8	32	10	17	6	0	25	8
2	Sausage McMuffin	35	42	15	33	10	17	8	0	25	10
3	Sausage McMuffin with Egg	43	52	95	36	10	17	15	0	30	15
4	Sausage McMuffin with Egg Whites	35	42	16	37	10	17	6	0	25	10

```
In [186]: least_order.describe()
```

Out[186]:

	Total Fat (% Daily Value)	Saturated Fat (% Daily Value)	Cholesterol (% Daily Value)	Sodium (% Daily Value)	Carbohydrates (% Daily Value)	Dietary Fiber (% Daily Value)	Vitamin A (% Daily Value)	Vitamin C (% Daily Value)	Calcium (% Daily Value)	Iron (% Daily Value)
count	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000	260.000000
mean	21.815385	29.965385	18.392308	20.676923	15.780769	6.530769	13.426923	8.534615	20.973077	7.734615
std	21.885199	26.639209	29.091653	24.034954	9.419544	6.307057	24.366381	26.345542	17.019953	8.723265
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	3.750000	4.750000	2.000000	4.750000	10.000000	0.000000	2.000000	0.000000	6.000000	0.000000
50%	17.000000	24.000000	11.000000	8.000000	15.000000	5.000000	8.000000	0.000000	20.000000	4.000000
75%	35.000000	48.000000	21.250000	36.250000	20.000000	10.000000	15.000000	4.000000	30.000000	15.000000
max	182.000000	102.000000	192.000000	150.000000	47.000000	28.000000	170.000000	240.000000	70.000000	40.000000

Kemudian menggunakan nilai max pada describe sebagai acuan dalam memilih items apa yang dapat memenuhi nutrisi keseharian kita. Hasil yang didapatkan sebagai berikut:

```

In [190]: #Memasukan nilai maksimum
max_index=[]
for i in least_order.describe().columns:
    max_index.extend(least_order[least_order[i]==least_order.describe().loc['max',i]].index.tolist())
#Menghilangkan duplikasi nilai
index_set = set(max_index)
#Dibuat menjadi list
max_list = list(index_set)

max_list

Out[190]: [32, 89, 34, 135, 82, 84, 246, 88, 249, 253, 31]

```

Data tersebut masih belum sampai disitu, masih ada tahap filter Kembali mencari item mana saja yang cocok untuk memenuhi nutrisi harian. Dengan diasumsikan untuk memenuhi nutrisi harian membutuhkan minimum 90 %Daily Value pada setiap kolom. Sehingga hasil yang didapat sebagai berikut:

```

In [193]: from itertools import combinations

for i in combinations(max_data.index,6):
    table = least_order.iloc[list(i),:]
    l = []
    for n in max_data.columns[1:]:
        m = table[n].sum()
        l.append(m)
    #Asumsi nilai maksimum per items sebanyak 90
    if all(l>=90*np.ones(10)):
        print(i)
        break

(32, 89, 34, 135, 82, 84)

```



```
In [194]: minimum_order = least_order.iloc[list(i),:]
          minimum_order
```

Out[194]:

	Item	Total Fat (% Daily Value)	Saturated Fat (% Daily Value)	Cholesterol (% Daily Value)	Sodium (% Daily Value)	Carbohydrates (% Daily Value)	Dietary Fiber (% Daily Value)	Vitamin A (% Daily Value)	Vitamin C (% Daily Value)	Calcium (% Daily Value)	Iron (% Daily Value)
32	Big Breakfast with Hotcakes (Large Biscuit)	93	100	192	94	39	28	15	2	30	40
89	Premium Southwest Salad with Grilled Chicken	13	13	23	28	9	28	170	30	15	15
34	Big Breakfast with Hotcakes and Egg Whites (La...	77	81	19	95	38	28	4	2	25	30
135	Minute Maid Orange Juice (Large)	0	0	0	0	22	0	0	240	4	0
82	Chicken McNuggets (40 piece)	182	101	89	150	39	24	0	15	8	25
84	Premium Bacon Ranch Salad (without Chicken)	11	18	9	13	3	12	170	30	15	6

Berdasarkan hasil analisis kombinasi makanan yang memungkinkan dapat memenuhi nutrisi harian ada 6 kombinasi, diantaranya adalah:

- 1) Big Breakfast with Hotcakes (Large Biscuit)
- 2) Premium Southwest Salad with Grilled Chicken
- 3) Big Breakfast with Hotcakes and Egg White (Large Biscuit)
- 4) Minute Maid Orange Juice (Large)
- 5) Chicken McNuggets (40 piece)
- 6) Premium Bacon Ranch Salad (Without Chicken)

Ke-6 kombinasi makanan diatas adalah saran berdasarkan nilai nutrisi yang terkandung, namun belum tentu makanan yang ada disebutkan diatas adalah kombinasi makanan yang sehat. Kita harus bisa cermat dan berhati-hati dengan yang disebut “overnutrition” dari sebageian nutrisi seperti sodium, lemak, kolesterol, dan nutrisi lainnya yang masing-masing memiliki batasan. Berikut adalah manfaat yang dapat dirasakan jika mengkonsumsi ke-6 kombinasi tersebut:

```
In [195]: for s in least_order.describe().columns:  
          print(s,': ', minimum_order[s].sum())
```

```
Total Fat (% Daily Value) : 376  
Saturated Fat (% Daily Value) : 313  
Cholesterol (% Daily Value) : 332  
Sodium (% Daily Value) : 380  
Carbohydrates (% Daily Value) : 150  
Dietary Fiber (% Daily Value) : 120  
Vitamin A (% Daily Value) : 359  
Vitamin C (% Daily Value) : 319  
Calcium (% Daily Value) : 97  
Iron (% Daily Value) : 116
```