

Reviewer's feedback v.1

Halo Tamara

Greetings from Chamdani

Terima kasih telah mengirimkan project kamu dengan ini kamu sudah memulai sebuah langkah yang luar biasa saat ini. Disini saya akan mereview proyek kamu ya.

Saya akan memberikan beberapa komentar dan feedback seperti dibawah ini, **mohon jangan dipindah, dirubah, maupun dihapus ya :)**.

Mohon diperhatikan bahwa apabila ada temuan atau kesalahan yang sama namun tidak ada komentar perbaikan disana, itu berarti kamu masih tetap harus memperbaikinya yaa.

Komentar yang saya berikan akan muncul dalam warna hijau, kuning, atau pun merah seperti ini:

Chamdani's comment v.*

Bagus, semua berjalan lancar.

Chamdani's comment v.*

Ada beberapa catatan.

Chamdani's comment v.*

Perlu beberapa perbaikan.

Chamdani's info v.1

Catatan umum:

- Fungsi `print` tidak diperlukan untuk mencetak suatu nilai
- Lebih baik tulis kode pada bagian akhir cell code untuk menampilkan output tanpa `print`
- Perintah soal yang ada dalam tanda kurung siku [...] sebaiknya dihilangkan
- Sangat disarankan untuk menggunakan cell markdown untuk memberikan deskripsi temuan atau kesimpulan.

- Tidak disarankan menggunakan cell code untuk memberikan penjelasan / deskripsi.
- Untuk menghindari galat pada notebook project kamu. Batasi jumlah output yang dikeluarkan setiap kode dengan batas max 10 baris data.

INGATLAH

- Project kamu tidak akan bisa diterima bila kamu masih memiliki feedback berwarna merah.
- Kamu hanya memiliki 3 kali kesempatan iterasi submission disetiap projectnya
- Kamu dapat memperbaiki dan menyelesaikan feedback **pertama** dari saya tidak lebih dari 1*24 jam. *Untuk membantu saya mengetahui apakah ada kesulitan dari feedback project kamu*
- Jika merasa kesulitan kamu dapat menuliskan responmu atas feedback saya menggunakan blok markdown warna biru yaa.

Kamu dapat menjawab saya dengan menggunakan blok berwarna biru seperti ini

Student answer

Siap kak.

Chamdani's comment v.1

Ingatlah kamu harus selalu memberikan penjelasan terkait apa yang kamu lakukan, apa yang kamu temukan, dan apa yang menjadi kesimpulan sementaramu **disetiap prosesnya**. Gunakanlah markdown cell bukan print atau comment dalam cell code

Ingatlah saya tidak akan mengingatkan ini disetiap bagian pada project, jadi saya akan memintamu sekali untuk semua bagian.

Untuk membantumu menjelaskan proses dan insight yang kamu temukan secara lebih rinci dan menggunakan kalimat yang naratif saya telah membuat daftar pertanyaan yang akan sangat membantu.

Jelaskan apa yang kamu temukan:

- apakah ada nilai yang aneh atau anomali?
- bagaimana nilai anomali itu bisa terjadi?
- apa saja yang kamu lakukan untuk mengatasi anomali tersebut?
- apa yang melandasimu untuk melakukan hal tersebut?
- apa insight awal yang kamu harapkan?
- apa hasil sebenarnya?

Chamdani's comment v.1

Contoh deskripsi / kesimpulan yang baik perhatikan DO and DON'T berikut yaa

DO

Iakukan hal ini Memberikan penjelasan yang naratif dan terstruktur seperti kalanya kamu bercerita tentang data kepada orang khalayak umum. Lakukan seperti seorang profesional yang memberikan penjelasan yang mudah dicerna, menarik dan mengandung informasi baru. Disini juga dijelaskan rencana untuk langkah selanjutnya

2.1 Kesimpulan

Setiap baris dalam tabel menyimpan data terkait trek lagu yang diputar. Sejumlah kolom menyimpan data yang mendeskripsikan trek itu sendiri: judul trek, artis, dan genre. Sisanya menyimpan data terkait informasi pengguna: kota asal mereka, waktu mereka memutar trek lagu tersebut.

Jelas bahwa data yang kita miliki cukup untuk menguji hipotesis. Meski demikian, kita memiliki nilai-nilai yang hilang.

Untuk melanjutkan analisis, kita perlu melakukan pra-pemrosesan data terlebih dahulu.

Dari informasi diatas kita mendapatkan beberapa gambaran:

- Dataset kita berjumlah **16715** baris dan **11** kolom.
- Beberapa kolom terdapat *missing value* seperti kolom `Name`, `Year_of_Release`, `Genre`, `Critic_Score`, `User_Score`, dan `Rating`.
- Kita akan mengganti register dari nama-nama kolom menjadi huruf kecil untuk mempermudah dalam melakukan analisis.
- Kolom `Tahun` seharusnya tidak didefinisikan sebagai `object` melainkan `datetime`, dan kolom `user_score` tipe datanya seharusnya `float` bukan `object`.

DON'T

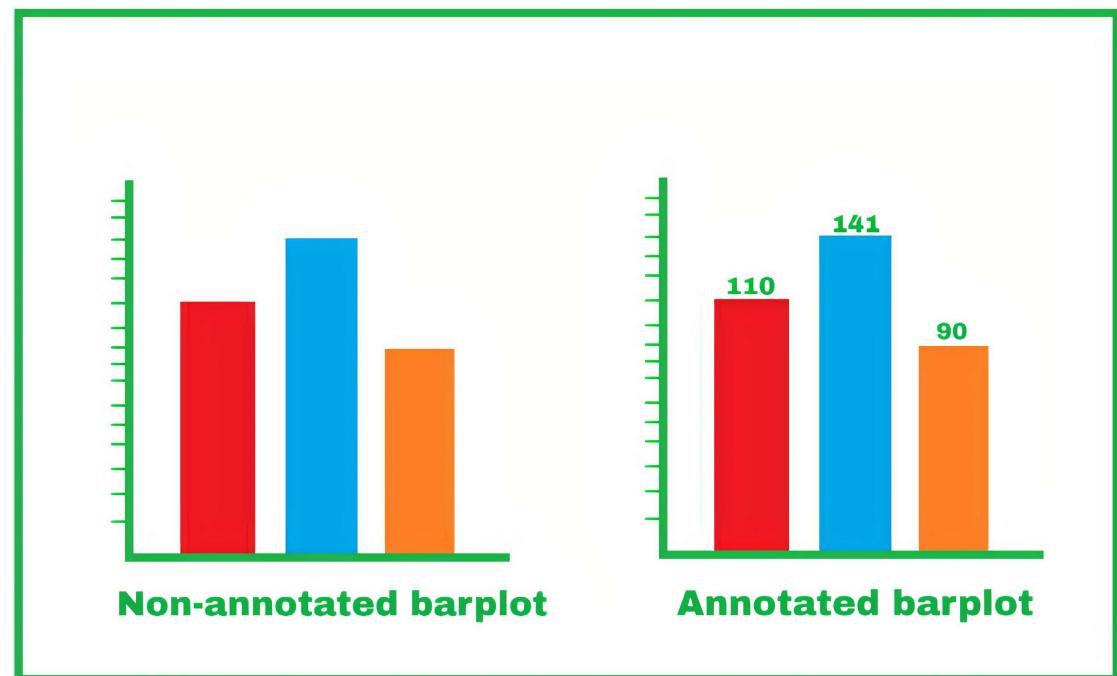
JANGAN Iakukan hal ini Memberikan penjelasan yang terlalu sederhana dan tidak mengandung informasi baru. Dan sebenarnya penjelasan tersebut sudah digambarkan dalam output kode. Ini sangat kurang baik untuk dilakukan. Akan lebih baik jika kamu memberikan penjelasan secara naratif dan menarik seperti halnya profesional data scientist atau analyst



```
In [4]: data.isna().sum()
Out[4]: Name
```

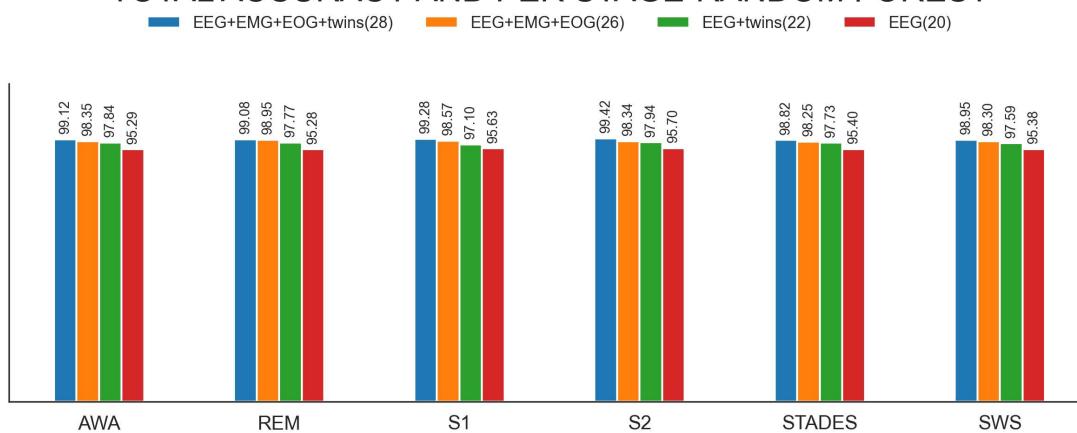
Chamdani's info v.1

Sebagai saran untuk project selanjutnya agar graph plot kamu lebih informatif dan lebih baik lagi kamu bisa menambahkan anotasi seperti pada contoh gambar berikut:



Dan terkadang grafik kita memiliki posisi gambar yang cukup berdekatan oleh karena itu kita bisa membuat sedikit rotasi pada label value pada setiap plot seperti contoh berikut:

TOTAL ACCURACY AND PER STAGE-RANDOM FOREST



Bacalah referensi berikut ini yaa:

- <https://stackoverflow.com/questions/39519609/annotate-bars-with-values-on-pandas-on-seaborn-factorplot-bar-plot>
(<https://stackoverflow.com/questions/39519609/annotate-bars-with-values-on-pandas-on-seaborn-factorplot-bar-plot>)
- <https://www.geeksforgeeks.org/how-to-annotate-bars-in-barplot-with-matplotlib-in-python/> (<https://www.geeksforgeeks.org/how-to-annotate-bars-in-barplot-with-matplotlib-in-python/>)

Contoh kode: kamu bisa memperbaiki kode ini untuk kamu gunakan pada project selanjutnya

```
#membuat visualisi untuk game rilis pertahun
plt.figure(figsize=(25,20)) # contoh kode saya perbesar figure size nya
splot = sns.barplot(data=year_group, x='year', y='name') # assign plot kedalam object
#contoh kode anotasi mulai
for g in splot.patches:
    splot.annotate(format(g.get_height(), '.1f'),
                   (g.get_x() + g.get_width() / 2., g.get_height()),
                   ha = 'center', va = 'center', # posisi label value vertical dan horizontal berada pada center plot
                   xytext = (0, 20), rotation=45, # memutar label value pada plot 45 derajat dan memberi jarak 20 pixel dari plot
                   textcoords = 'offset points')
#contoh kode anotasi selesai
plt.xticks(rotation = 45) # memutar label x axis 45 derajat
plt.show()
```

Chamdani's comment v.1

Silahkan gunakan fungsi berikut untuk melakukan pengisian data null dengan median. Kamu bisa mengganti dengan mean, median atau yang lain sesuai dengan analismu

```
| def fill_nan_median(dataframe, agg_column, value_column):
    grouped_values = dataframe.groupby(agg_column)[value_column].median().reset_index()
    size = len(grouped_values)
    for i in range(size):
        group = grouped_values[agg_column][i]
        value = grouped_values[value_column][i]
        dataframe.loc[(dataframe[agg_column]==group) & (dataframe[value_column].isna()), value_column] = value
    return dataframe
```

Gunakan fungsi yang saya sarankan tadi dengan cara seperti ini.

```
fill_nan_median(data, "model", "model_year")
```

Chamdani's comment v.1

INGATLAH pada setiap project, selain deskripsi proyek kamu juga harus memberikan pengantar proyek, daftar isi, dan tujuan proyek. Tepat seperti yang telah kami contohkan pada project pertama tentang Y-music.

Contents

- [Introduction](#)
- [Stage 1. Data overview](#)
 - [Conclusions](#)
- [Stage 2. Data preprocessing](#)
 - [2.1 Header style](#)
 - [2.2 Missing values](#)
 - [2.3 Duplicates](#)
 - [2.4 Conclusions](#)
- [Stage 3. Testing the hypotheses](#)
 - [3.1 Hypothesis 1: user activity in the two cities](#)
 - [3.2 Hypothesis 2: music preferences on Monday and Friday](#)
 - [3.3 Hypothesis 3: genre preferences in Springfield and Shelbyville](#)
- [Findings](#)

1 Introduction

Whenever we're doing research, we need to formulate hypotheses that we can then test. Sometimes we accept these hypotheses; other times, we reject them. To make the right decisions, a business must be able to understand whether or not it's making the right assumptions.

In this project, you'll compare the music preferences of the cities of Springfield and Shelbyville. You'll study real Yandex.Music data to test the hypotheses below and compare user behavior for these two cities.

1.1 Goal:

Test three hypotheses:

1. User activity differs depending on the day of the week and from city to city.
2. On Monday mornings, Springfield and Shelbyville residents listen to different genres. This is also true for Friday evenings.
3. Springfield and Shelbyville listeners have different preferences. In Springfield, they prefer pop, while Shelbyville has more rap fans.

1.2 Stages

Data on user behavior is stored in the file `/datasets/music_project_en.csv`. There is no information about the quality of the data, so you will need to explore it before testing the hypotheses.

First, you'll evaluate the quality of the data and see whether its issues are significant. Then, during data preprocessing, you will try to account for the most critical problems.

General Result

General feedback v.1 [Major Improvement Needed]

Terima kasih banyak telah mengirimkan proyek Anda!

Secara keseluruhan, proyek Anda **SANGAT** mengesankan. Namun saya telah menemukan beberapa hal kecil yang bisa menjadi saran perbaikan agar proyek kamu menjadi lebih baik lagi.

Saya telah meninggalkan komentar yang dapat membantu Anda dalam hal tersebut. Hal ini tidak berarti bahwa Anda telah melakukan sesuatu yang salah. Hal ini hanya berisi beberapa saran tentang bagaimana pekerjaan Anda dapat kembangkan lebih lanjut. Anda masih memiliki kesempatan untuk memperbaikinya dan jangan lupa untuk kembali kepada saya setelah perbaikan anda selesai.

Jika ada sesuatu yang membuat Anda bingung atau tidak mengerti. Tolong, jangan ragu untuk menghubungi tim mentor atau instruktur kamu yaa

kamu juga bisa menghubungi saya melalui reply feedback di notebook ini yaa

| TETAP SEMANGAT!! |

General feedback v.2 [Major Improvement Needed]

Terima kasih banyak telah mengirimkan proyek Anda!

Secara keseluruhan, proyek Anda **SANGAT** mengesankan. Namun saya telah menemukan beberapa hal kecil yang bisa menjadi saran perbaikan agar proyek kamu menjadi lebih baik lagi.

Saya telah meninggalkan komentar yang dapat membantu Anda dalam hal tersebut. Hal ini tidak berarti bahwa Anda telah melakukan sesuatu yang salah. Hal ini hanya berisi beberapa saran tentang bagaimana pekerjaan Anda dapat kembangkan lebih lanjut. Anda masih memiliki kesempatan untuk memperbaikinya dan jangan lupa untuk kembali kepada saya setelah perbaikan anda selesai.

Jika ada sesuatu yang membuat Anda bingung atau tidak mengerti. Tolong, jangan ragu untuk menghubungi tim mentor atau instruktur kamu yaa

kamu juga bisa menghubungi saya melalui reply feedback di notebook ini yaa

TETAP SEMANGAT!!

General feedback v.3 Project Accepted

Terima kasih banyak telah mengirimkan proyek Anda!

Secara keseluruhan, proyek Anda **SANGAT** mengesankan. Semoga apa yang kamu pelajari dalam project ini dapat membantu meningkatkan kemampuanmu. Pertahankan apa yang sudah bagus, dan tingkatkan apa yang menurutmu masih perlu ditingkatkan. Semoga berhasil pada project selanjutnya yaa :)

TETAP SEMANGAT!!

Y.Music



Konten

- [Pendahuluan](#)
- [Tahap 1. Ikhtisar Data](#)
 - [Kesimpulan](#)
- [Tahap 2. Pra-pemrosesan data](#)
 - [2.1 Gaya Penulisan Judul](#)

- [2.2 Nilai-Nilai yang Hilang](#)
- [2.3 Duplikat](#)
- [2.4 Kesimpulan](#)
- [Tahap 3. Pengujian Hipotesis](#)
 - [3.1 Hipotesis 1: Aktivitas pengguna di kedua kota](#)
 - [3.2 Hipotesis 2: Preferensi musik pada hari Senin dan Jumat](#)
 - [3.3 Hipotesis 3: Preferensi genre di kota Springfield dan Shelbyville](#)
- [Temuan](#)

Pendahuluan

Setiap kali kita melakukan analisis, kita perlu merumuskan beberapa hipotesis yang perlu kita uji lebih lanjut. Terkadang, pengujian yang kita lakukan membawa kita untuk menerima hipotesis tersebut. Namun di lain waktu, kita juga perlu menolaknya. Untuk membuat keputusan yang tepat dalam bisnis, kita harus memahami apakah asumsi yang kita buat sudah tepat atau belum.

Dalam proyek kali ini, kamu akan membandingkan preferensi musik dari pendengar di kota Springfield dan Shelbyville. Kamu akan meninjau data sungguhan dari Y.Music untuk menguji beberapa hipotesis di bawah ini dan membandingkan perilaku pengguna pada kedua kota.

Tujuan:

Menguji tiga hipotesis:

1. Aktivitas pengguna berbeda-beda tergantung pada hari dan kotanya.
2. Pada hari Senin pagi, penduduk Springfield dan Shelbyville mendengarkan genre yang berbeda. Hal ini juga berlaku untuk hari Jumat malam.
3. Pendengar di kota Springfield dan Shelbyville memiliki preferensi yang berbeda. Di Springfield, para pengguna lebih suka musik pop, sementara di Shelbyville musik rap memiliki lebih banyak penggemar.

Tahapan

Data terkait perilaku pengguna disimpan dalam file `/datasets/music_project_en.csv`. Tidak ada informasi terkait kualitas data tersebut, karena itu kamu perlu memeriksanya terlebih dahulu sebelum menguji hipotesis.

Pertama-tama, kamu akan mengevaluasi kualitas data dan melihat apakah masalahnya signifikan. Kemudian, selama pra-pemrosesan data, kamu akan mencoba mengatasi masalah yang paling serius.

Proyek ini terdiri dari tiga tahap:

1. Tinjauan data
2. Pra-pemrosesan data
3. Pengujian hipotesis

[Kembali ke Konten](#)

Chamdani's comment v.1

Bagus, semua berjalan lancar. Ingatlah kamu perlu memberikan pengantar setiap mengawali proyek seperti contoh di atas. Karena setelah proyek pengenalan ini kamu akan diminta melakukannya sendiri yaa

Tahap 1. Tinjauan data

Buka data terkait Y.Music, lalu pelajarilah data tersebut.

Kamu akan membutuhkan library Pandas , oleh karena itu silakan mengimpornya.

In [1]: `# mengimpor Pandas
import pandas as pd`

Chamdani's comment v.1

Hebat, Kerja bagus! Import package selalu dilakukan diawal notebook dan dilakukan selaki saja

Baca file `music_project_en.csv` dari folder `/datasets/` lalu simpan file tersebut di variabel `df` :

In [2]:

```
df = pd.read_csv('/datasets/music_project_en.csv') # Load dataset  
df.describe() # menampilkan informasi statistik sederhana
```

Out[2]:

	userID	Track	artist	genre	City	time	Day
count	65079	63736	57512	63881	65079	65079	65079
unique	41748	39666	37806	268	2	20392	3
top	A8AE9169	Brand	Kartveli	pop	Springfield	21:51:22	Friday
freq	76	136	136	8850	45360	14	23149

**Chamdani's comment v.1**

Hebat, Kerja bagus! Namun kamu tidak memerlukan kode baris pertama. Disini kamu hanya perlu load dataset dan tidak perlu import package lagi

Chamdani's comment v.2

Kode yang benar seperti ini

```
import pandas as pd # ini kode untuk import package SEHARUSNYA DIH  
APUS

df = pd.read_csv('/datasets/music_project_en.csv') # Load dataset

df.describe() # menampilkan informasi statistik sederhana
```

Chamdani's comment v.3

Hebat, Kerja bagus!

Tampilkan 10 baris pertama tabel:

In [3]: `# Memperoleh 10 baris pertama dari tabel df
df.head(10)`

Out[3]:

	userID	Track	artist	genre	City	time	Day
0	FFB692EC	Kamigata To Boots	The Mass Missile	rock	Shelbyville	20:28:33	Wednesday
1	55204538	Delayed Because of Accident	Andreas Rönnberg	rock	Springfield	14:07:09	Friday
2	20EC38	Funiculi funiculà	Mario Lanza	pop	Shelbyville	20:58:07	Wednesday
3	A3DD03C9	Dragons in the Sunset	Fire + Ice	folk	Shelbyville	08:37:09	Monday
4	E2DC1FAE	Soul People	Space Echo	dance	Springfield	08:34:34	Monday
5	842029A1	Chains	Obladaet	rusrap	Shelbyville	13:09:41	Friday
6	4CB90AA5	True	Roman Messer	dance	Springfield	13:00:07	Wednesday
7	F03E1C1F	Feeling This Way	Polina Griffith	dance	Springfield	20:47:49	Wednesday
8	8FA1D3BE	L'estate	Julia Dalia	ruspop	Springfield	09:17:40	Friday
9	E772D5C0	Pessimist	NaN	dance	Shelbyville	21:20:49	Wednesday

Dapatkan informasi umum tentang tabel dengan satu perintah:

Chamdani's comment v.1

Hebat, Kerja bagus!

```
In [4]: # Memperoleh informasi umum tentang data yang tersedia di df
df.info() # menampilkan informasi umum dataset

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 65079 entries, 0 to 65078
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   userID      65079 non-null   object  
 1   Track       63736 non-null   object  
 2   artist      57512 non-null   object  
 3   genre       63881 non-null   object  
 4   City        65079 non-null   object  
 5   time        65079 non-null   object  
 6   Day         65079 non-null   object  
dtypes: object(7)
memory usage: 3.5+ MB
```

Chamdani's comment v.1

Hebat, Kerja bagus! Namun tidak perlu load ulang datasetnya yaa

Chamdani's comment v.2

Kode yang benar seperti ini

```
df = pd.read_csv('/datasets/music_project_en.csv') # Load dataset
SEHARUSNYA DIHAPUS
```

```
df.info() # menampilkan informasi umum dataset
```

Chamdani's comment v.3

Hebat, Kerja bagus!

Tabel ini berisi tujuh kolom. Semua kolom memiliki tipe data yang sama, yaitu: `object`.

Berdasarkan dokumentasi:

- 'userID' — ID pengguna
- 'Track' — judul lagu
- 'artist' — nama artis
- 'genre'
- 'City' — kota asal pengguna
- 'time' — waktu saat lagu tersebut diputar



- 'Day' — hari dalam seminggu

Kita dapat melihat tiga masalah dengan gaya penulisan nama kolom:

1. Beberapa nama ditulis dalam huruf besar, beberapa dalam huruf kecil.
2. Beberapa nama menggunakan spasi.
3. Carilah sendiri masalah ketiga dalam gaya penulisan nama kolom dan jelaskan masalah tersebut di sini .

Kita juga dapat melihat bahwa terdapat jumlah nilai yang berbeda antar kolom. Hal ini menunjukkan bahwa data yang kita miliki mengandung nilai yang hilang

Kesimpulan

Setiap baris dalam tabel menyimpan data terkait trek lagu yang diputar. Beberapa kolom menyimpan data yang mendeskripsikan trek itu sendiri: judul lagu, artis, dan genre. Sisanya menyimpan data terkait informasi pengguna: kota asal mereka, waktu mereka memutar trek lagu tersebut.

Jelas bahwa data yang kita miliki cukup untuk menguji hipotesis. Sayangnya, terdapat sejumlah nilai yang hilang.

Untuk melanjutkan analisis, kita perlu melakukan pra-pemrosesan data terlebih dahulu.

Chamdani's comment v.1

Bagus, semua berjalan lancar. Ingatlah setiap sesi kamu perlu menyusun deskripsi dan kesimpulan dari setiap langkah dan temuanmu untuk melatih kamu menjadi profesional di bidang data

[Kembali ke Daftar Isi](#)

Tahap 2. Pra-pemrosesan data

Perbaiki format pada judul kolom dan atasi nilai yang hilang. Kemudian, periksa apakah datamu mengandung duplikat.



Gaya penulisan judul

Tampilkan judul kolom:

```
In [5]: # List yang memuat nama-nama kolom pada tabel df
(df.columns)
```

```
Out[5]: Index(['userID', 'Track', 'artist', 'genre', 'City', 'time', 'Day'], dt
ype='object')
```

Chamdani's comment v.1

Bagus, semua berjalan lancar. Namun sebaiknya tidak perlu menggunakan fungsi `print` untuk mencetak nilai dalam notebook. karena kamu bisa melakukannya dengan cara meletakkan kode pada bagian akhir cell code untuk menampilkan isinya.

**catatan: ini berlaku untuk bagian-bagian selanjutnya,
kamu perlu memperbaiki jika ada kesalahan yang sama**

Chamdani's comment v.2

Hebat, Kerja bagus!

Ubah nama kolom sesuai dengan aturan gaya penulisan yang baik:

- Jika nama kolom terdiri dari beberapa kata, gunakan `snake_case`
- Semua karakter harus menggunakan huruf kecil
- Hapus spasi

```
In [6]: # mengganti nama kolom
# Mengubah nama kolom sesuai dengan aturan gaya penulisan yang baik
df = df.rename(columns={
    'userID': 'user_id',
    'Track': 'track',
    'artist': 'artist',
    'genre': 'genre',
    'City': 'city',
    'time': 'time',
    'Day': 'day'
})
# Menampilkan nama kolom setelah perubahan
```

**Chamdani's comment v.1**

Kode baris terakhir tidak diperlukan silahkan dihapus

Chamdani's comment v.2

Periksa hasilnya. Tampilkan nama kolom sekali lagi:

In [7]: `# mengecek hasilmu: tampilkan sekali lagi list yang memuat nama-nama kolom
(df.columns)`

Out[7]: `Index(['user_id', 'track', 'artist', 'genre', 'city', 'time', 'day'], dtype='object')`

Chamdani's comment v.1

ingat komentar tentang print yaa dan tidak perlu menggunakan method `.tolist()`

Chamdani's comment v.2

Hebat, Kerja bagus!

[Kembali ke Daftar Isi](#)

Nilai-nilai yang hilang

Pertama, temukan jumlah nilai yang hilang dalam tabel. Untuk melakukannya, gunakan dua metode Pandas :

In [8]: `# menghitung nilai yang hilang
Menggunakan isna() untuk mendeteksi nilai-nilai yang hilang dan sum() untuk n
missing_values = df.isna().sum()

print(f"Jumlah nilai yang hilang: {missing_values}")`

```
Jumlah nilai yang hilang: user_id      0
track        1343
artist       7567
genre        1198
city          0
time          0
day           0
dtype: int64
```



Chamdani's comment v.1

ingat komentar tentang print statement dan kode yang benar adalah `df.isna().sum()`
lainnya bisa dihapus yaa

Chamdani's comment v.2

Hebat, Kerja bagus!

Tidak semua nilai yang hilang berpengaruh terhadap penelitianmu. Misalnya, nilai yang hilang dalam kolom `track` dan `artist` tidak begitu penting. Kamu cukup menggantinya dengan penanda yang jelas. Akan tetapi, nilai yang hilang dalam kolom '`genre`' dapat memengaruhi perbandingan preferensi musik di kota Springfield dan Shelbyville. Dalam kehidupan nyata, sangatlah berguna untuk mempelajari alasan hilangnya data tersebut dan mencoba memperbaikinya. Sayangnya, kita belum memiliki kesempatan tersebut dalam proyek ini. Oleh karena itu, kamu harus:

- Mengisi nilai yang hilang dengan penanda
- Mengevaluasi seberapa besar pengaruh nilai yang hilang terhadap perhitunganmu

Ganti nilai yang hilang pada kolom '`track`' , '`artist`' , dan '`genre`' dengan string '`unknown`' . Untuk melakukannya, buat list yang bernama `columns_to_replace` , terapkan loop `for` pada list tersebut, dan ganti nilai yang hilang di setiap kolom:

```
In [9]: # menerapkan Loop terhadap nama-nama kolom dan mengganti nilai yang hilang dengan 'unknown'
# Daftar nama kolom yang ingin diperiksa
columns_to_replace = ['track', 'artist', 'genre']

# Mengganti nilai yang hilang dengan 'unknown' menggunakan Loop for
for column in columns_to_replace:
    df[column].fillna('unknown', inplace=True)
```

Chamdani's comment v.1

Hebat, Kerja bagus!

Pastikan tidak ada tabel lagi yang berisi nilai yang hilang. Hitung kembali nilai yang hilang.



```
In [10]: # Menghitung nilai yang hilang setelah penggantian
missing_values_after_replace = df.isna().sum()

print(f"Jumlah nilai yang hilang setelah penggantian: {missing_values_after_replace}")
# menghitung nilai yang hilang
```

```
Jumlah nilai yang hilang setelah penggantian: user_id      0
track        0
artist       0
genre        0
city         0
time         0
day          0
dtype: int64
```

Chamdani's comment v.1

ingat komentar tentang print statement dan kode yang benar adalah `df.isna().sum()` lainnya bisa dihapus yaa

Chamdani's comment v.2

Hebat, Kerja bagus!

[Kembali ke Daftar Isi](#)

Duplikat

Temukan jumlah duplikat eksplisit dalam tabel menggunakan satu perintah:

```
In [11]: # menghitung duplikat eksplisit
df.duplicated().sum()
```

Out[11]: 3826

Chamdani's comment v.1

ingat komentar tentang print statement dan kode yang benar adalah `df.duplicated().sum()` lainnya bisa dihapus yaa

Chamdani's comment v.2

Panggil salah satu metode Pandas untuk menghapus duplikat eksplisit:

In [12]:

```
# menghapus duplikat eksplisit
df = df.drop_duplicates().reset_index(drop=True)
```

Chamdani's comment v.1

Silahkan perbaiki kode mu seperti ini df =
df.drop_duplicates().reset_index(drop=True)

Chamdani's comment v.2

Kode baris terakhir bisa dihapus

Chamdani's comment v.3

Hebat, Kerja bagus!

Hitung duplikat eksplisit sekali lagi untuk memastikan bahwa kamu telah berhasil menghapus semuanya:

In [13]:

```
# memeriksa duplikat
df.duplicated().sum()
```

Out[13]: 0

Chamdani's comment v.1

ingat komentar tentang print statement dan kode yang benar adalah
df.duplicated().sum() lainnya bisa dihapus yaa



Chamdani's comment v.2

Hebat, Kerja bagus!

Sekarang, hapus duplikat implisit di kolom genre . Sebagai contoh, penulisan suatu nama genre dengan cara yang berbeda merupakan contoh dari duplikat implisit. Kesalahan seperti ini juga akan memengaruhi hasil analisismu.

Tampilkan list yang memuat nama unik genre, lalu urutkan list tersebut berdasarkan abjad. Untuk melakukannya:

- Ambil kolom DataFrame yang diinginkan
- Terapkan metode pengurutan pada kolom tersebut\n",
- Untuk kolom yang telah diurutkan, panggil metode yang akan menghasilkan semua nilai unik kolom

In [14]: `df['genre'].sort_values().unique()`

...

Chamdani's comment v.1

ingat komentar tentang print yaa dan kode yang lebih tepat adalah seperti ini
`df['genre'].sort_values().unique()`

Chamdani's comment v.2

Penulisan kode keliru tolong lebih teliti yaa

Chamdani's comment v.3

Hebat, Kerja bagus!

Lihat baik-baik list yang telah ditampilkan untuk menemukan duplikat implisit dari genre hiphop . Duplikat tersebut bisa berupa nama yang ditulis secara tidak tepat atau nama alternatif dari genre yang sama.

Kamu akan melihat duplikat implisit berikut:

- hip
- hop
- hip-hop

Untuk menghapusnya, gunakan fungsi `replace_wrong_genres()` dengan dua parameter:

- `wrong_genres=` — list dengan duplikat yang ingin diganti
- `correct_genre=` — string dengan nilai yang benar

Fungsi tersebut harus mengoreksi nama dalam kolom 'genre' dari tabel df , mengganti setiap nilai dari list wrong genres dengan nilai dari correct genre .

```
In [15]: # masukkan fungsi yang mengganti duplikat implisit
def replace_wrong_genres(wrong_genres, correct_genre):
    for wrong_genre in wrong_genres:
        df['genre'] = df['genre'].replace(wrong_genre, correct_genre)
```

Chamdani's comment v.1

Kode kurang tepat bisa dihapus dan kode yang benar seperti ini **Kamu seharusnya hanya mendefinisikan fungsi replace_wrong_genres disini**

```
def replace_wrong_genres(wrong_genres, correct_genre):
    for wrong_genre in wrong_genres:
        df['genre'] = df['genre'].replace(wrong_genre, correct_gen
re)
```

Chamdani's comment v.2

Hebat, Kerja bagus!

Panggil replace_wrong_genres() dan teruskan argumen kepada fungsi tersebut, sehingga ia dapat menghapus duplikat implisit (hip , hop , dan hip-hop) dan menggantinya dengan hiphop :

```
In [16]: # menghapus duplikat implisit
duplicates = ['hip', 'hop', 'hip-hop']
genre = 'hiphop'
replace_wrong_genres(duplicates, genre)
```

Chamdani's comment v.1

Selanjutnya **Kamu seharusnya hanya perlu memanggil fungsi replace_wrong_genres disini**

```
duplicates = ['hip', 'hop', 'hip-hop']
genre = 'hiphop'
replace_wrong_genres(duplicates, genre)
```

Chamdani's comment v.2

Hebat, Kerja bagus!

Pastikan bahwa nilai yang terduplikasi telah dihapus. Tampilkan list nilai unik dari kolom 'genre' :

```
In [17]: # memeriksa duplikat implisit  
df['genre'].sort_values().unique()
```

```
Out[17]: array(['acid', 'acoustic', 'action', 'adult', 'africa', 'afrikaans',
   'alternative', 'ambient', 'americana', 'animated', 'anime',
   'arabesk', 'arabic', 'arena', 'argentinetango', 'art', 'audiobook',
   'avantgarde', 'axé', 'baile', 'balkan', 'beats', 'bigroom',
   'black', 'bluegrass', 'blues', 'bollywood', 'bossa', 'brazilian',
   'breakbeat', 'breaks', 'broadway', 'cantautorí', 'cantopop',
   'canzone', 'caribbean', 'caucasian', 'celtic', 'chamber',
   'children', 'chill', 'chinese', 'choral', 'christian', 'christmas',
   'classical', 'classicmetal', 'club', 'colombian', 'comedy',
   'conjazz', 'contemporary', 'country', 'cuban', 'dance',
   'dancehall', 'dancepop', 'dark', 'death', 'deep', 'deutschrock',
   'deutschspr', 'dirty', 'disco', 'dnb', 'documentary', 'downbeat',
   'downtempo', 'drum', 'dub', 'dubstep', 'eastern', 'easy',
   'electronic', 'electropop', 'emo', 'entehno', 'epicmetal',
   'estrada', 'ethnic', 'eurofolk', 'european', 'experimental',
   'extrememetal', 'fado', 'film', 'fitness', 'flamenco', 'folk',
   'folklore', 'folkmetal', 'folkrock', 'folktronica', 'forró',
   'frankreich', 'französisch', 'french', 'funk', 'future', 'gangsta',
   'garage', 'german', 'ghazal', 'gitarre', 'glitch', 'gospel',
```

Chamdani's comment v.1

ingat komentar tentang print yaa dan kode yang lebih tepat adalah seperti ini

```
df['genre'].sort_values().unique()
```

Chamdani's comment v.2

Hebat, Kerja bagus!

[Kembali ke Daftar Isi](#)

Kesimpulan

Kita telah mendekripsi tiga masalah dalam data kita:

- Gaya penulisan judul yang salah
 - Nilai yang hilang
 - Duplikat eksplisit dan implisit

Sekarang, nama-nama kolom telah dibersihkan untuk mempermudah pemrosesan tabel. Semua nilai yang hilang pun telah diganti dengan 'unknown'. Meski demikian, kita masih harus melihat apakah nilai yang hilang dalam kolom 'genre' akan memengaruhi perhitungan kita.

Ketidaaan duplikat akan membuat hasil yang kita dapat menjadi lebih tepat dan lebih mudah dipahami.

.....
[Kembali ke Daftar Isi](#)

Tahap 3. Pengujian hipotesis

Hipotesis 1: membandingkan perilaku pengguna di dua kota

Menurut hipotesis pertama, pengguna dari kota Springfield dan Shelbyville memiliki perbedaan perilaku dalam mendengarkan musik. Pengujian ini menggunakan data yang diambil dari tiga hari dalam seminggu: Senin, Rabu, dan Jumat.

- Bagilah pengguna ke dalam beberapa kelompok berdasarkan kota.
- Bandingkan berapa banyak trek lagu yang dimainkan oleh setiap kelompok pada hari Senin, Rabu, dan Jumat.

Lakukan setiap perhitungan secara terpisah agar kamu bisa berlatih.

Evaluasi aktivitas pengguna di setiap kota. Kelompokkan data berdasarkan kota dan temukan jumlah trek lagu yang diputar di setiap kelompok.



In [18]: df

Out[18]:

	user_id	track	artist	genre	city	time	day
0	FFB692EC	Kamigata To Boots	The Mass Missile	rock	Shelbyville	20:28:33	Wednesday
1	55204538	Delayed Because of Accident	Andreas Rönnberg	rock	Springfield	14:07:09	Friday
2	20EC38	Funiculì funiculà	Mario Lanza	pop	Shelbyville	20:58:07	Wednesday
3	A3DD03C9	Dragons in the Sunset	Fire + Ice	folk	Shelbyville	08:37:09	Monday
4	E2DC1FAE	Soul People	Space Echo	dance	Springfield	08:34:34	Monday
...
61248	729CBB09	My Name	McLean	rnb	Springfield	13:32:28	Wednesday
61249	D08D4A55	Maybe One Day (feat. Black Spade)	Blu & Exile	hiphop	Shelbyville	10:00:00	Monday
61250	C5E3A0D5	Jalopiina	unknown	industrial	Springfield	20:09:26	Friday
61251	321D0506	Freight Train	Chas McDevitt	rock	Springfield	21:43:59	Friday
61252	3A64EF84	Tell Me Sweet Little Lies	Monica Lopez	country	Springfield	21:59:46	Friday

61253 rows × 7 columns

In [19]:

```
# Buat DataFrame dengan data pengguna dan jumlah trek Lagu yang diputar
df.groupby('city')['track'].count().reset_index()
```

Out[19]:

	city	track
0	Shelbyville	18512
1	Springfield	42741

Chamdani's comment v.1

Disini kamu tidak perlu import package lagi, kamu hanya perlu menampilkan kolom city dan track saja yaa. dan sebaiknya tidak perlu menggunakan fungsi print untuk mencetak nilai. Perhatikan kode dibawah ini

```
df.groupby('city')['track'].count().reset_index()
```

**Chamdani's comment v.2**

Chamdani's comment v.3

Hebat, Kerja bagus!

Pengguna dari kota Springfield memutar lebih banyak trek lagu daripada pengguna dari kota Shelbyville. Akan tetapi, hal ini tidak serta-merta mengisyaratkan bahwa warga Springfield lebih sering mendengarkan musik. Kota ini memang lebih besar, dan terdapat lebih banyak pengguna. Jadi, ini merupakan hal yang wajar.

Sekarang, kelompokkan data berdasarkan hari dan temukan jumlah trek lagu yang diputar pada hari Senin, Rabu, dan Jumat.

In [20]:

```
# Menghitung trek Lagu yang diputar pada masing-masing hari
# Buat DataFrame dengan data pengguna dan jumlah trek Lagu yang diputar
df.groupby('day')[['track']].count().reset_index()
```

Out[20]:

	day	track
0	Friday	21840
1	Monday	21354
2	Wednesday	18059

Chamdani's comment v.1

Disini kamu tidak perlu import package lagi, kamu hanya perlu menampilkan kolom `day` dan `track` saja yaa. dan sebaiknya tidak perlu menggunakan fungsi `print` untuk mencetak nilai. Perhatikan kode dibawah ini

```
df.groupby('day')[['track']].count().reset_index()
```

Chamdani's comment v.2

Kode import package tidak diperlukan silahakn dihapus

**Chamdani's comment v.3**

Hebat, Kerja bagus!

Rabu adalah hari yang paling "tenang" secara keseluruhan. Namun jika kita mempertimbangkan kedua kota secara terpisah, kita mungkin akan mendapatkan kesimpulan yang berbeda.

Kamu telah melihat cara kerja pengelompokan berdasarkan kota atau hari. Sekarang, tuliskan sebuah fungsi yang akan mengelompokkan data berdasarkan kota dan hari.

Buat fungsi `number_tracks()` untuk menghitung jumlah trek lagu yang diputar untuk hari dan kota tertentu. Fungsi tersebut akan membutuhkan dua parameter:

- nama hari dalam seminggu
- nama kota

Dalam fungsi yang kita buat, gunakan variabel untuk menyimpan baris dari tabel asli, di mana:

- Nilai kolom 'day' sama dengan parameter `day` \n",
- Nilai kolom 'city' sama dengan parameter `city`

Terapkan pemfilteran berurutan dengan pengindeksan logika.

Kemudian, hitung nilai kolom 'user_id' pada tabel yang dihasilkan. Simpan hasilnya ke dalam variabel baru. Kembalikan variabel ini dari fungsi.

In [21]:

`df`

Out[21]:

	user_id	track	artist	genre	city	time	day
0	FFB692EC	Kamigata To Boots	The Mass Missile	rock	Shelbyville	20:28:33	Wednesday
1	55204538	Delayed Because of Accident	Andreas Rönnerberg	rock	Springfield	14:07:09	Friday
2	20EC38	Funiculi funiculà	Mario Lanza	pop	Shelbyville	20:58:07	Wednesday
3	A3DD03C9	Dragons in the Sunset	Fire + Ice	folk	Shelbyville	08:37:09	Monday
4	E2DC1FAE	Soul People	Space Echo	dance	Springfield	08:34:34	Monday
...
61248	729CBB09	My Name	McLean	rnb	Springfield	13:32:28	Wednesday
61249	D08D4A55	Maybe One Day (feat. Black Spade)	Blu & Exile	hiphop	Shelbyville	10:00:00	Monday
61250	C5E3A0D5	Jalopiina	unknown	industrial	Springfield	20:09:26	Friday
61251	321D0506	Freight Train	Chas McDevitt	rock	Springfield	21:43:59	Friday
61252	3A64EF84	Tell Me Sweet Little Lies	Monica Lopez	country	Springfield	21:59:46	Friday

61253 rows × 7 columns

In [22]: # Buat DataFrame dengan data pengguna dan jumlah trek Lagu yang diputar

```
def number_tracks(day, city):
    # Filter data berdasarkan kota dan hari
    filtered_data = df[(df['day'] == day) & (df['city'] == city)]

    # Hitung jumlah trek Lagu yang diputar
    total_tracks = filtered_data['track'].count()

    return total_tracks

# Contoh pemanggilan fungsi
jumlah_trek_senin_springfield = number_tracks('Monday', 'Springfield')
print(f"Jumlah trek lagu yang diputar di Springfield pada hari Senin: {jumlah_trek_senin_springfield}")

jumlah_trek_rabu_shelbyville = number_tracks('Wednesday', 'Shelbyville')
print(f"Jumlah trek lagu yang diputar di Shelbyville pada hari Rabu: {jumlah_trek_rabu_shelbyville}")
```

Jumlah trek lagu yang diputar di Springfield pada hari Senin: 15740
 Jumlah trek lagu yang diputar di Shelbyville pada hari Rabu: 7003

Chamdani's comment v.1

Disini kamu tidak perlu import package lagi, Kode kamu bisa hapus semua karena tidak sesuai dengan perintah soal untuk contoh benarnya Perhatikan kode dibawah ini

```
def number_track(day,city):
    track_list = df[(df['day'] == day) & (df['city'] == city)]
    track_list_count = track_list['user_id'].count()
    return (track_list_count)
```

Chamdani's comment v.2

Kode import package tidak diperlukan silahkan dihapus. Tolong lebih teliti dan hati2 dalam mengikuti saran review yaa.

Chamdani's comment v.3

Hebat, Kerja bagus!

Panggil `number_tracks()` sebanyak enam kali dan ubahlah nilai parameternya pada masing-masing panggilan, sehingga kamu bisa mengambil data dari kedua kota untuk masing-masing hari (Senin, Rabu, dan Jumat).

In [23]: `# jumlah lagu yang diputar di Springfield pada hari Senin
number_tracks('Monday', 'Springfield')`

Out[23]: 15740

Chamdani's comment v.1

Ada beberapa catatan disini

- ingat komentar tentang print yaa
- tidak perlu membuat nama2 hari sendiri
- gunakan informasi yang ada dalam dataset df
- Gunakan keyword value yang sesuai dengan value dalam dataset
- Hindari menggunakan keyword value yang tidak sesuai dengan dataset seperti penggunaan **Senin** yang keliru seharusnya menggunakan **Monday**
- Kode yang benar tampak seperti ini `number_tracks('Monday', 'Springfield')`

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya

Chamdani's comment v.3

Hebat, Kerja bagus!

In [24]: `# jumlah lagu yang diputar di Shelbyville pada hari Senin
jumlah_lagu_monday_shelbyville = number_tracks('Monday', 'Shelbyville')
print(f"Jumlah lagu yang diputar di Shelbyville pada hari Senin: {jumlah_lagu_m`

Jumlah lagu yang diputar di Shelbyville pada hari Senin: 5614

Chamdani's comment v.1

Ada beberapa catatan disini

- ingat komentar tentang print yaa
- tidak perlu membuat nama2 hari sendiri
- gunakan informasi yang ada dalam dataset df
- Gunakan keyword value yang sesuai dengan value dalam dataset
- Hindari menggunakan keyword value yang tidak sesuai dengan dataset seperti penggunaan **Senin** yang keliru seharusnya menggunakan **Monday**

- Kode yang benar tampak seperti ini `number_track('Monday', 'Shelbyville')`

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya

Chamdani's comment v.3

Hebat, Kerja bagus!

```
In [25]: # jumlah lagu yang diputar di Springfield pada hari Rabu
jumlah_lagu_wednesday_springfield = number_tracks('Wednesday', 'Springfield')
print(f"Jumlah lagu yang diputar di Springfield pada hari Rabu: {jumlah_lagu_we
```

Jumlah lagu yang diputar di Springfield pada hari Rabu: 11056

Chamdani's comment v.1

Ada beberapa catatan disini

- ingat komentar tentang print yaa
- tidak perlu membuat nama2 hari sendiri
- gunakan informasi yang ada dalam dataset `df`
- Gunakan keyword `value` yang sesuai dengan `value` dalam dataset
- Hindari menggunakan keyword `value` yang tidak sesuai dengan dataset seperti penggunaan `**Senin**` yang keliru seharusnya menggunakan `**Monday**`
- Kode yang benar tampak seperti ini `number_track('Wednesday', 'Springfield')`

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya



Chamdani's comment v.3

Hebat, Kerja bagus!

```
In [26]: # jumlah lagu yang diputar di Shelbyville pada hari Rabu
jumlah_lagu_wednesday_shelbyville = number_tracks('Wednesday', 'Shelbyville')
print(f"Jumlah lagu yang diputar di Shelbyville pada hari Rabu: {jumlah_lagu_wednesday_shelbyville}")
```

Jumlah lagu yang diputar di Shelbyville pada hari Rabu: 7003

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya

Chamdani's comment v.1

Ada beberapa catatan disini

- ingat komentar tentang print yaa
- tidak perlu membuat nama2 hari sendiri
- gunakan informasi yang ada dalam dataset df
- Gunakan keyword value yang sesuai dengan value dalam dataset
- Hindari menggunakan keyword value yang tidak sesuai dengan dataset seperti penggunaan **Senin** yang keliru seharusnya menggunakan **Monday**
- Kode yang benar tampak seperti ini number_track('Wednesday', 'Shelbyville')

Chamdani's comment v.3

Hebat, Kerja bagus!

```
In [27]: # jumlah lagu yang diputar di Springfield pada hari Jumat
jumlah_lagu_friday_springfield = number_tracks('Friday', 'Springfield')
print(f"Jumlah lagu yang diputar di Springfield pada hari Jumat: {jumlah_lagu_friday_springfield}")
```

Jumlah lagu yang diputar di Springfield pada hari Jumat: 15945

Chamdani's comment v.1

Ada beberapa catatan disini

- ingat komentar tentang print yaa
- tidak perlu membuat nama2 hari sendiri
- gunakan informasi yang ada dalam dataset df
- Gunakan keyword value yang sesuai dengan value dalam dataset

- Hindari menggunakan keyword value yang tidak sesuai dengan dataset seperti penggunaan **Senin** yang keliru seharusnya menggunakan **Monday**

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya

Chamdani's comment v.3

Hebat, Kerja bagus!

In [28]: `# jumlah lagu yang diputar di Shelbyville pada hari Jumat
jumlah_lagu_friday_shelbyville = number_tracks('Friday', 'Shelbyville')
print(f"Jumlah lagu yang diputar di Shelbyville pada hari Jumat: {jumlah_lagu_friday_shelbyville}")`

Jumlah lagu yang diputar di Shelbyville pada hari Jumat: 5895

Chamdani's comment v.1

Ada beberapa catatan disini

- ingat komentar tentang print yaa
- tidak perlu membuat nama2 hari sendiri
- gunakan informasi yang ada dalam dataset df
- Gunakan keyword value yang sesuai dengan value dalam dataset
- Hindari menggunakan keyword value yang tidak sesuai dengan dataset seperti penggunaan **Senin** yang keliru seharusnya menggunakan **Monday**
- Kode yang benar tampak seperti ini `number_track('Friday', 'Shelbyville')`

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya



Chamdani's comment v.3

Hebat, Kerja bagus!

Gunakan pd.DataFrame untuk membuat sebuah tabel, yang

- Nama kolomnya adalah: ['city', 'monday', 'wednesday', 'friday']
- Datanya adalah hasil yang kamu dapatkan dari number_tracks()

In [29]: # tabel yang memuat hasil
import pandas as pd

```
# Panggil number_tracks() untuk mendapatkan data
jumlah_lagu_senin_springfield = number_tracks('Monday', 'Springfield')
jumlah_lagu_senin_shelbyville = number_tracks('Monday', 'Shelbyville')
jumlah_lagu_rabu_springfield = number_tracks('Wednesday', 'Springfield')
jumlah_lagu_rabu_shelbyville = number_tracks('Wednesday', 'Shelbyville')
jumlah_lagu_jumat_springfield = number_tracks('Friday', 'Springfield')
jumlah_lagu_jumat_shelbyville = number_tracks('Friday', 'Shelbyville')

# Membuat DataFrame
data = [['Springfield', number_tracks('Monday', 'Springfield'), number_tracks('Wednesday', 'Springfield'), number_tracks('Friday', 'Springfield')], ['Shelbyville', number_tracks('Monday', 'Shelbyville'), number_tracks('Wednesday', 'Shelbyville'), number_tracks('Friday', 'Shelbyville')]]
header = ['city', 'Monday', 'Wednesday', 'Friday']

pd.DataFrame(data=data, columns=header)
```

Out[29]:

	city	Monday	Wednesday	Friday
0	Springfield	15740	11056	15945
1	Shelbyville	5614	7003	5895

Chamdani's comment v.1

Ada beberapa catatan disini

- ingat komentar tentang print
- Disini kamu tidak perlu import package lagi
- tidak perlu membuat nama2 hari sendiri
- gunakan informasi yang ada dalam dataset df
- Gunakan keyword value yang sesuai dengan value dalam dataset
- Hindari menggunakan keyword value yang tidak sesuai dengan dataset seperti penggunaan **Senin** yang keliru seharusnya menggunakan **Monday**
- Kode yang benar tampak seperti ini

```
# tabel yang memuat hasil
data = [['Springfield', number_track('Monday', 'Springfield'), numb
          ...]
```

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya

Chamdani's comment v.3

Hebat, Kerja bagus!

Kesimpulan

Data yang kamu dapatkan ternyata berhasil mengungkapkan beberapa perbedaan perilaku pengguna:

- Di kota Springfield, jumlah trek lagu yang diputar mencapai puncaknya pada hari Senin dan Jumat, sedangkan pada hari Rabu terjadi penurunan aktivitas.
- Di kota Shelbyville, sebaliknya, pengguna lebih banyak mendengarkan musik pada hari Rabu. Aktivitas pengguna pada hari Senin dan Jumat lebih sedikit.

Dengan demikian, dapat disimpulkan bahwa hipotesis pertama tampaknya benar.

[Kembali ke Daftar Isi](#)

Hipotesis 2: musik di awal dan akhir minggu

Menurut hipotesis kedua, pada hari Senin pagi dan Jumat malam, warga Springfield mendengarkan genre musik yang berbeda dari yang dinikmati warga Shelbyville.

Dapatkan tabel (pastikan nama tabel gabungannya cocok dengan DataFrame yang diberikan dalam dua blok kode di bawah):

- Untuk Springfield — spr_general
- Untuk Shelbyville — shel_general



```
In [30]: # mendapatkan tabel spr_general dari baris df,
# yang nilai dari kolom 'city'-nya adalah 'Springfield'
spr_general = df[df['city'] == 'Springfield']
```

Chamdani's comment v.1

Hebat, Kerja bagus! namun disini kamu tidak perlu menggunakan method `.copy()` karena kamu tidak akan menyalin data dari `df` melainkan hanya menyimpan data `df` dengan kriteria tertentu dalam variable baru

Chamdani's comment v.2

Hebat, Kerja bagus!

In [31]:

```
# mendapatkan shel_general dari baris df,
# yang nilai dari kolom 'city'-nya adalah 'Shelbyville'
shel_general = df[df['city'] == 'Shelbyville']
```

Chamdani's comment v.1

Hebat, Kerja bagus! namun disini kamu tidak perlu menggunakan method `.copy()` karena kamu tidak akan menyalin data dari `df` melainkan hanya menyimpan data `df` dengan kriteria tertentu dalam variable baru

Chamdani's comment v.2

Hebat, Kerja bagus!

Buat fungsi `genre_weekday()` dengan empat parameter:

- Sebuah tabel untuk data
- Nama hari
- Stempel waktu awal, dalam format 'hh:mm'
- Stempel waktu akhir, dalam format 'hh:mm'

Fungsi tersebut harus menghasilkan informasi tentang 15 genre paling populer pada hari tertentu dalam periode antara dua stempel waktu.



```
In [32]: # Mendeklarasikan fungsi genre_weekday() dengan parameter day=, time1=, dan time2=, dan memberikan informasi tentang genre yang paling populer pada hari dan waktu tertentu

# 1) Buat agar variabel genre_df menyimpan baris yang memenuhi beberapa kondisi
#     - nilai pada kolom 'day' sama dengan nilai argumen day=
#     - nilai pada kolom 'time' lebih besar dari nilai argumen time1=
#     - nilai pada kolom 'time' lebih kecil dari nilai argumen time2=
#     Gunakan pemfilteran berurutan dengan pengindeksan logika.

# 2) Kelompokkan genre_df berdasarkan kolom 'genre', ambil salah satu kolomnya, kemudian gunakan metode count() untuk menemukan jumlah entri untuk masing-masing genre yang diwakili; simpan Series yang dihasilkan ke dalam variabel genre_df_count

# 3) Urutkan genre_df_count dalam urutan menurun berdasarkan frekuensi dan simpangan standar ke dalam variabel genre_df_sorted

# 4) Hasilkan sebuah objek Series dengan 15 nilai genre_df_sorted pertama - 15 nilai genre_df_sorted terakhir (pada hari tertentu, dalam jangka waktu tertentu)

# tulis fungsi Anda di sini
import pandas as pd

def genre_weekday(data, day, time1, time2):
    # pemfilteran berurutan
    # genre_df hanya akan menyimpan baris df yang day-nya sama dengan day
    genre_df = data[data['day'] == day]

    # genre_df hanya akan menyimpan baris df yang time-nya lebih kecil dari time1
    genre_df = genre_df[genre_df['time'] > time1]

    # genre_df hanya akan menyimpan baris df yang time-nya lebih besar dari time2
    genre_df = genre_df[genre_df['time'] < time2]

    # kelompokkan DataFrame yang telah difilter berdasarkan kolom dengan nama genre
    genre_df_grouped = genre_df.groupby('genre')['user_id'].count()

    # kita akan mengurutkan hasilnya dalam urutan menurun (sehingga genre yang paling populer diatas)
    genre_df_sorted = genre_df_grouped.sort_values(ascending=False)

    # kita akan menghasilkan objek Series yang menyimpan 15 genre paling populer
    return genre_df_sorted[:15]
```

Chamdani's comment v.1

Hebat, Kerja bagus!

Bandingkan hasil dari fungsi genre_weekday() untuk Springfield dan Shelbyville pada hari Senin pagi (dari pukul 07.00 hingga 11.00) dan pada hari Jumat malam (dari pukul 17:00 hingga 23:00):

```
In [33]: # memanggil fungsi untuk Senin pagi di Springfield (gunakan spr_general alih-alih df)
spr_general = df[df['city'] == 'Springfield']
genre_weekday(spr_general, 'Monday', '07:00', '11:00')
```

```
Out[33]: genre
pop           781
dance          549
electronic     480
rock           474
hiphop          286
ruspop          186
world           181
rusrap          175
alternative    164
unknown         161
classical       157
metal            120
jazz             100
folk              97
soundtrack       95
Name: user_id, dtype: int64
```

Chamdani's comment v.1

Kode keliru, kode yang benar hanya seperti ini.

```
genre_weekday(spr_general, 'Monday', '07:00', '11:00')
```

Ingatlah bahwa kamu hanya perlu memanggil variable, fungsi atau attribut yang telah kamu definisikan sebelumnya tidak perlu membuat yang baru jika tidak ada perubahan data disana

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya

Chamdani's comment v.3

Hebat, Kerja bagus!



```
In [34]: # memanggil fungsi untuk Senin pagi di Shelbyville (gunakan shel_general alih-alih df)
shel_general = df[df['city'] == 'Springfield'].copy()
genre_weekday(shel_general, 'Monday', '07:00', '11:00')
```

```
Out[34]: genre
pop           781
dance          549
electronic     480
rock           474
hiphop          286
ruspop          186
world           181
rusrap          175
alternative    164
unknown         161
classical       157
metal            120
jazz             100
folk              97
soundtrack       95
Name: user_id, dtype: int64
```

Chamdani's comment v.1

Kode keliru, kode yang benar hanya seperti ini.

```
genre_weekday(shel_general, 'Monday', '07:00', '11:00')
```

Ingatlah bahwa kamu hanya perlu memanggil variable, fungsi atau attribut yang telah kamu definisikan sebelumnya tidak perlu membuat yang baru jika tidak ada perubahan data disana

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya

Chamdani's comment v.3

Hebat, Kerja bagus!



```
In [35]: # memanggil fungsi untuk Jumat malam di Springfield\nspr_general = df[df['city'] == 'Springfield'].copy()\ngenre_weekday(spr_general, 'Friday', '17:00', '23:00')
```

```
Out[35]: genre\npop           713\nrock          517\ndance          495\nelectronic     482\nhiphop         273\nworld          208\nruspop         170\nclassical      163\nalternative    163\nrusrap          142\njazz            111\nunknown         110\nsoundtrack      105\nrnb              90\nmetal             88\nName: user_id, dtype: int64
```

Chamdani's comment v.1

Kode keliru, kode yang benar hanya seperti ini.

```
genre_weekday(spr_general, 'Friday', '17:00', '23:00')
```

Ingatlah bahwa kamu hanya perlu memanggil variable, fungsi atau attribut yang telah kamu definisikan sebelumnya tidak perlu membuat yang baru jika tidak ada perubahan data disana

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya

Chamdani's comment v.3

Hebat, Kerja bagus!



```
In [36]: # memanggil fungsi untuk Jumat malam di Shelbyville  
shel_general = df[df['city'] == 'Springfield'].copy()  
genre_weekday(shel_general, 'Friday', '17:00', '23:00')
```

```
Out[36]: genre  
pop           713  
rock          517  
dance          495  
electronic     482  
hiphop         273  
world          208  
ruspop         170  
classical      163  
alternative    163  
rusrap          142  
jazz            111  
unknown         110  
soundtrack      105  
rnb              90  
metal             88  
Name: user_id, dtype: int64
```

Chamdani's comment v.1

Kode keliru, kode yang benar hanya seperti ini.

```
genre_weekday(shel_general, 'Friday', '17:00', '23:00')
```

Ingatlah bahwa kamu hanya perlu memanggil variable, fungsi atau attribut yang telah kamu definisikan sebelumnya tidak perlu membuat yang baru jika tidak ada perubahan data disana

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya

Chamdani's comment v.3

Hebat, Kerja bagus!

Kesimpulan

Setelah membandingkan 15 genre teratas pada hari Senin pagi, kita dapat menarik kesimpulan berikut:

1. Pengguna dari Springfield dan Shelbyville mendengarkan musik dengan genre yang sama.
Lima genre teratas dari kedua kota sama, hanya genre rock dan elektronik yang bertukar tempat.
2. Di Springfield, jumlah nilai yang hilang ternyata sangat besar, sehingga nilai 'unknown' pun berada di urutan ke-10. Artinya, nilai-nilai yang hilang mencakup proporsi data yang cukup besar, sehingga fakta ini bisa dijadikan dasar untuk mempertanyakan reliabilitas kesimpulan kami.

Untuk hari Jumat malam, situasinya juga serupa. Genre individual cukup bervariasi, tetapi secara keseluruhan, 15 besar genre untuk kedua kota sama.

Dengan demikian, hipotesis kedua terbukti benar sebagian:

- Pengguna mendengarkan musik yang sama di awal dan di akhir minggu.
- Tidak ada perbedaan yang mencolok antara Springfield dan Shelbyville. Di kedua kota tersebut, pop adalah genre yang paling populer.

Meski demikian, signifikansi jumlah nilai yang hilang membuat hasil ini patut dipertanyakan. Di Springfield, terdapat begitu banyak nilai hilang yang memengaruhi hasil 15 genre teratas kita.

....

[Kembali ke Daftar Isi](#)

Hipotesis 3: preferensi genre di kota Springfield dan Shelbyville

Hipotesis: pendengar di kota Shelbyville menyukai musik rap, sementara pendengar di kota Springfield lebih menyukai pop.

Kelompokkan tabel `spr_general` berdasarkan genre dan temukan jumlah trek lagu yang dimainkan untuk setiap genre dengan metode `count()`. Kemudian, urutkan hasilnya dalam urutan menurun dan simpanlah ke `spr_genres`.

In [37]: `# Kelompokkan tabel spr_general berdasarkan genre dan hitung jumlah trek Lagu
spr_genres = spr_general.groupby('genre')[['track']].count().sort_values(ascending=False)`

Chamdani's comment v.1

Hebat, Kerja bagus! namun kode kurang optimal yaa. Silahkan optimalkan kode seperti ini

```
spr_genres = spr_general.groupby('genre')[['track']].count().sort_values(ascending=False)
```



Chamdani's comment v.2

Hebat, Kerja bagus!

Tampilkan 10 baris pertama dari `spr_genres` :

In [38]: `# menampilkan 10 baris pertama dari spr_genres
spr_genres.head(10)`

Out[38]: genre
pop 5892
dance 4435
rock 3965
electronic 3786
hiphop 2096
classical 1616
world 1432
alternative 1379
ruspop 1372
rusrap 1161
Name: track, dtype: int64

Chamdani's comment v.1

Disini kamu seharunya menampilkan 10 data baris pertama pada dataframe `spr_genres` kamu bisa menggunakan method `.head(n)` untuk menampilkan baris pertama data sebanyak `n` baris

Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya

Chamdani's comment v.3

Hebat, Kerja bagus!

Sekarang, lakukan hal yang sama pada data dari Shelbyville.

Kelompokkan tabel `shel_general` berdasarkan genre dan temukan jumlah trek lagu yang dimainkan untuk setiap genre. Kemudian, urutkan hasilnya dalam urutan menurun dan simpan hasil tersebut ke tabel `shel_genres` :

In [39]: # dalam satu baris: kelompokkan tabel shel_general berdasarkan kolom 'genre',
 shel_genres = shel_general.groupby('genre')[['track']].count().sort_values(ascending=True)
 # hitung nilai kolom 'genre' dalam pengelompokan menggunakan count(),
 # urutkan Series yang dihasilkan dalam urutan menurun dan simpanlah ke shel_genres

Chamdani's comment v.1

Disini kamu seharunya membuat dataframe shel_genres Silahkan optimalkan kode seperti ini
 shel_genres = shel_general.groupby('genre')[['track']].count().sort_values(ascending=False)

Chamdani's comment v.2

Hebat, Kerja bagus!

Tampilkan 10 baris pertama dari shel_genres :

In [40]: # menampilkan 10 baris pertama dari shel_genres
 shel_genres.head(10)

Out[40]: genre
 pop 5892
 dance 4435
 rock 3965
 electronic 3786
 hiphop 2096
 classical 1616
 world 1432
 alternative 1379
 ruspop 1372
 rusrap 1161
 Name: track, dtype: int64

Chamdani's comment v.1

Disini kamu seharunya menampilkan 10 data baris pertama pada dataframe shel_genres kamu bisa menggunakan method .head(n) untuk menampilkan baris pertama data sebanyak n baris



Chamdani's comment v.2

Tolong lebih teliti dan hati2 dalam mengikuti saran review Masih banyak hal yang kamu lewatkan di review sebelumnya

Chamdani's comment v.3

Hebat, Kerja bagus!

Kesimpulan

Hipotesis ini terbukti benar sebagian:

- Musik pop adalah genre yang paling populer di Springfield, seperti yang kita perkirakan.
- Namun, musik pop ternyata sama populernya baik di Springfield maupun di Shelbyville, dan musik rap ternyata tidak masuk ke daftar 5 besar genre untuk kedua kota tersebut.

[Kembali ke Daftar Isi](#)

Temuan

Kita telah menguji tiga hipotesis berikut:

1. Aktivitas pengguna berbeda-beda tergantung pada hari dan kotanya.
2. Pada hari Senin pagi, penduduk Springfield dan Shelbyville mendengarkan genre yang berbeda. Hal ini juga berlaku untuk hari Jumat malam.
3. Pendengar di kota Springfield dan Shelbyville memiliki preferensi yang berbeda. Baik di Springfield maupun di Shelbyville, pengguna lebih menyukai musik pop.

Setelah menganalisis data yang tersedia, kita dapat menyimpulkan bahwa:

1. Aktivitas pengguna di Springfield dan Shelbyville bergantung pada hari dalam seminggu, meskipun kedua kota ini bervariasi dalam berbagai cara.

Hipotesis pertama dapat diterima sepenuhnya.

2. Preferensi musik tidak bervariasi secara signifikan sepanjang minggu di Springfield dan Shelbyville. Kita bisa melihat perbedaan kecil dalam urutan pada hari Senin, tetapi:

- Baik di Springfield maupun di Shelbyville, pengguna paling banyak mendengarkan musik pop.

Oleh karena itu, hipotesis ini tidak dapat kita terima. Penting juga untuk mengingat bahwa hasil yang diperoleh bisa saja berbeda seandainya kita tidak memiliki nilai yang hilang. 3. Ternyata, preferensi musik pengguna dari kota Springfield dan Shelbyville sangatlah mirip.

Hipotesis ketiga ditolak. Jika memang terdapat perbedaan preferensi, sayangnya hal ini tidak dapat kita ketahui dari data ini.

Catatan

Dalam proyek sesungguhnya, penelitian melibatkan pengujian hipotesis statistik, yang tentunya lebih tepat dan lebih bersifat kuantitatif. Perhatikan juga bahwa kamu tidak dapat selalu menarik kesimpulan tentang seluruh kota berdasarkan data dari satu sumber saja

Chamdani's comment v.1

Bagus, semua berjalan lancar. Selain deskripsi dan kesimpulan sementara pada akhir project kamu perlu membuat rangkuman temuan dan insight yang kamu dapatkan dari awal proses hingga proses selesai yaa

[Kembali ke Daftar Isi](#)

