

# LAPORAN HASIL DATA ANALYST

---

Nama Perusahaan : Ford Motor Company  
Nama Data Analyst : Shafira Nabilazzahra  
NPM : 11122367  
Jenjang/Program Studi : S1/Sistem Informasi  
Tanggal Laporan : 25 September 2025

Laporan Hasil Data Analyst ini Dibuat

Dalam Rangka Bukti Portofolio sebagai Data Analyst

---

## DAFTAR ISI

*[Gunakan fitur Table of Contents otomatis di Microsoft Word]*

DAFTAR ISI.....	2
1. PENDAHULUAN.....	3
1.1 Latar Belakang.....	3
1.2 Tujuan Analisis .....	3
2. IDENTIFIKASI KEBUTUHAN BISNIS .....	3
3. PENGUMPULAN & PERSIAPAN DATA.....	3
3.1 Sumber Data .....	4
3.2 Proses Praproses.....	5
4. ANALISIS DATA.....	14
4.1 Eksplorasi Variabel Utama .....	14
4.2 Visualisasi Data.....	15
5. TEMUAN & <i>INSIGHT</i> .....	18
6. PELUANG BISNIS.....	19
7. KESIMPULAN DAN REKOMENDASI .....	19
LAMPIRAN-LAMPIRAN.....	20

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Ford Motor Company ialah produsen kendaraan bermotor asal Amerika Serikat. Perusahaan ini menyediakan layanan jasa rental/penyewaan sepeda berbasis aplikasi. Awal diluncurkan pada 2013 aplikasi dinamakan Bay Area Bike Share, namun pada 2017 berganti nama menjadi Ford GoBike. Sistem ini dioperasikan oleh Motivate, sebuah perusahaan berbasis di New York yang menyediakan layanan berbagi, yaitu sepeda bisa dirental oleh banyak orang dalam waktu yang berbeda. Saat ini, Ford Motor Company menyediakan 2.500 sepeda dan 290 stasiun di San Francisco, San Jose dan Pantai Timur Bay.

Step harus dilakukan pelanggan untuk merental sepeda dari aplikasi Ford GoBike. Pelanggan bisa pergi ke stasiun sepeda, setelah menemukan sepeda yang ingin dirental, pelanggan bisa membuka kuncinya melalui aplikasi Ford GoBike. Pelanggan boleh meninggalkan sepeda di stasiun sepeda manapun yang berbasis di sekitar kota. Terdapat dua jenis pengguna aplikasi layanan Ford GoBike, yaitu *customers* dan *subscribers*. *Subscriber* ialah pengguna yang sering menggunakan paket berlangganan. Sedangkan *customer* ialah pengguna sesekali yang membeli satu kali perjalanan.

### 1.2 Tujuan Analisis

- Memahami pola antara penyewaan yang dilakukan oleh *customer* dan *subscriber*
- Mengidentifikasi durasi penggunaan aplikasi Ford GoBike dan stasiun yang paling sering didatangi oleh pengguna

## 2. IDENTIFIKASI KEBUTUHAN BISNIS

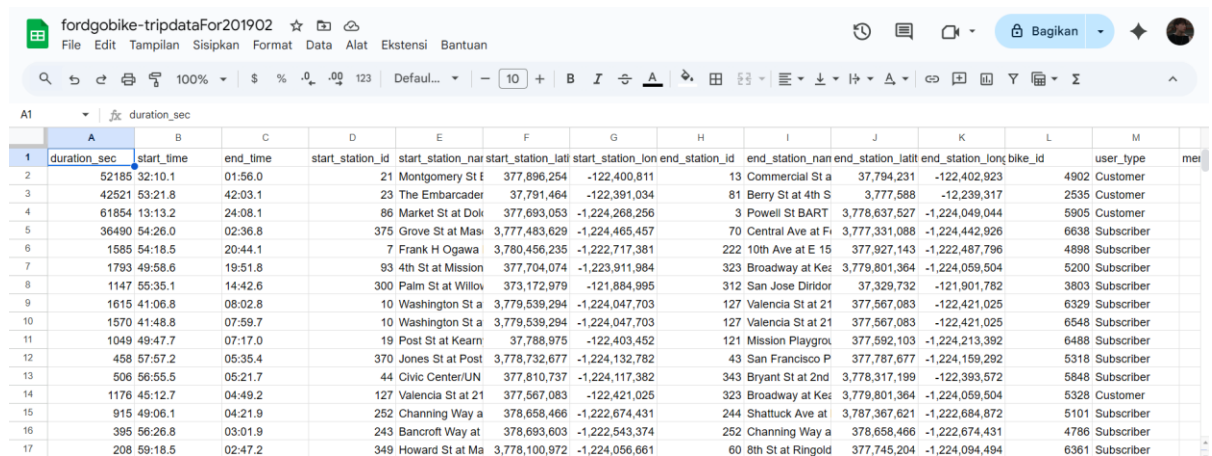
Dapat diuraikan:

- Bagaimana perbandingan frekuensi penyewaan sepeda antara *customer* dan *subscriber*?
- Apa stasiun yang sering didatangi pengguna dan memiliki peluang durasi penyewaan lebih besar?
- Siapakah yang menempuh perjalanan paling panjang?

### 3. PENGUMPULAN & PERSIAPAN DATA

#### 3.1 Sumber Data

Pengambilan data perusahaan Ford Motor Company menggunakan metode sekunder, yaitu mengambil data eksternal dari sumber lain. Sumber yang digunakan ialah <https://bit.ly/datasetFordGoBike>. Data yang tersedia menampilkan beberapa keterangan/fitur mengenai jasa rental sepeda menggunakan Ford GoBike. Keterangan pertama ialah `duration_sec` yaitu durasi waktu rental dalam rentang detik. Keterangan ke dua, `start_time` ialah data mengenai waktu tepat awal mula rental sepeda, dua digit pertama ialah menit, digit setelahnya adalah detik, adapun digit terakhir ialah keterangan milidetik. Keterangan ke tiga adalah akhir perjalanan dari waktu awal peminjaman, jika tertulis 01.56.0 maka waktu pemberhentian rental 1 menit 56 detik sejak menit awal rental. Keterangan selanjutnya mengenai stasiun, `start_station_id` ialah id stasiun mula rental sepeda, `start_station_name` adalah nama stasiun awal mula rental, `start_station_latitude` merupakan garis lintang dari stasiun awal mula rental, `start_station_longitude` adalah garis bujur tempat awal mula peminjaman, adapun penjelasan keterangan lain sama seperti `start/awal` mula rental, namun keterangan lainnya berupa akhir rental sepeda. Keterangan berikutnya ialah id sepeda yang digunakan oleh pengguna. Disebutkan apakah pelanggan yang mengakses termasuk ke dalam kategori *customer* dan *subscriber*. Keterangan selanjutnya mengenai pelanggan, yaitu `member_birth_year` yaitu tahun lahir pengguna dan `member_gender` yaitu jenis kelamin dari pengguna. Keterangan terakhir adalah `bike_share_for_all_trip` yaitu keterangan apakah sepeda sempat dibagi ke beberapa orang atau tidak.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name	end_station_latitude	end_station_longitude	bike_id	user_type	member_birth_year
2	52185	32:10.1	01:56.0	21	Montgomery St	37,786,254	-122,400,811	13	Commercial St	37,794,231	-122,402,923	4902	Customer	
3	42521	53:21.8	42:03.1	23	The Embarcadero	37,791,464	-122,391,034	81	Berry St at 4th S	3,777,588	-12,239,317	2535	Customer	
4	61854	13:13.2	24:08.1	86	Market St at Dol	37,769,053	-1,224,268,256	3	Powell St BART	3,778,637,527	-1,224,049,044	5905	Customer	
5	36490	54:26.0	02:36.8	375	Grove St at Mas	3,777,483,629	-1,224,465,457	70	Central Ave at Fi	3,777,331,088	-1,224,442,926	6638	Subscriber	
6	1585	54:18.5	20:44.1	7	Frank H Ogawa	3,780,456,235	-1,222,717,381	222	10th Ave at E 15	377,927,143	-1,222,487,796	4898	Subscriber	
7	1793	49:58.6	19:51.8	93	4th St at Mission	377,704,074	-1,223,911,984	323	Broadway at Kes	3,779,801,364	-1,224,059,504	5200	Subscriber	
8	1147	55:35.1	14:42.6	300	Palm St at Willov	373,172,979	-121,884,995	312	San Jose Diridori	37,329,732	-121,901,782	3803	Subscriber	
9	1615	41:06.8	08:02.8	10	Washington St a	3,779,539,294	-1,224,047,703	127	Valencia St at 21	377,567,083	-122,421,025	6329	Subscriber	
10	1570	41:48.8	07:59.7	10	Washington St a	3,779,539,294	-1,224,047,703	127	Valencia St at 21	377,567,083	-122,421,025	6548	Subscriber	
11	1049	49:47.7	07:17.0	19	Post St at Kearn	37,788,975	-1,224,403,452	121	Mission Playgro	377,592,103	-1,224,213,392	6488	Subscriber	
12	458	57:57.2	05:35.4	370	Jones St at Post	3,778,732,677	-1,224,132,782	43	San Francisco P	377,787,677	-1,224,159,292	5318	Subscriber	
13	506	56:55.5	05:21.7	44	Civic Center/UN	377,810,737	-1,224,117,382	343	Bryant St at 2nd	3,778,317,199	-1,222,393,572	5848	Subscriber	
14	1176	45:12.7	04:49.2	127	Valencia St at 21	377,567,083	-122,421,025	323	Broadway at Kes	3,779,801,364	-1,224,059,504	5328	Customer	
15	915	49:06.1	04:21.9	252	Channing Way a	378,658,466	-1,222,674,431	244	Shattuck Ave at	3,787,367,621	-1,222,684,872	5101	Subscriber	
16	395	56:26.8	03:01.9	243	Bancroft Way at	378,693,603	-1,222,543,374	252	Channing Way a	378,658,466	-1,222,674,431	4786	Subscriber	
17	208	59:18.5	02:47.2	349	Howard St at Ma	3,778,100,972	-1,224,056,661	60	8th St at Ringold	377,745,204	-1,224,094,494	6361	Subscriber	

### 3.2 Proses Praproses

- Pemeriksaan dan penghapusan data kosong

```
df.isnull()
```

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name	end_station_latitude	end_station_longitude	bike_id	user_type	member_birth_year	member_gender	bike_share_for_all_trip
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	True	True	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
103411	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
103412	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
103413	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
103414	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
103415	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

103416 rows x 16 columns

Untuk melakukan pemeriksaan missing values gunakan `df.isnull()`. Output ini berupa dataframe dengan nilai boolean, yaitu True jika sebuah sel bernilai kosong (missing value) dan False jika sel tersebut memiliki data. Dari tampilan terlihat bahwa hampir semua nilai adalah False, yang berarti sebagian besar data dalam dataset tidak memiliki kekosongan. Namun, ada beberapa nilai True pada baris tertentu yang menandakan adanya data yang hilang pada kolom tertentu.

<pre>miss = df.isnull().sum() miss</pre>	<pre>misspercent = (df.isnull().sum()/len(df))*100 misspercent</pre>																																																																				
<table><tr><th></th><th>0</th></tr><tr><td>duration_sec</td><td>0</td></tr><tr><td>start_time</td><td>0</td></tr><tr><td>end_time</td><td>0</td></tr><tr><td>start_station_id</td><td>197</td></tr><tr><td>start_station_name</td><td>197</td></tr><tr><td>start_station_latitude</td><td>0</td></tr><tr><td>start_station_longitude</td><td>0</td></tr><tr><td>end_station_id</td><td>197</td></tr><tr><td>end_station_name</td><td>197</td></tr><tr><td>end_station_latitude</td><td>0</td></tr><tr><td>end_station_longitude</td><td>0</td></tr><tr><td>bike_id</td><td>0</td></tr><tr><td>user_type</td><td>0</td></tr><tr><td>member_birth_year</td><td>8265</td></tr><tr><td>member_gender</td><td>8265</td></tr><tr><td>bike_share_for_all_trip</td><td>0</td></tr></table> <p>dtype: int64</p>		0	duration_sec	0	start_time	0	end_time	0	start_station_id	197	start_station_name	197	start_station_latitude	0	start_station_longitude	0	end_station_id	197	end_station_name	197	end_station_latitude	0	end_station_longitude	0	bike_id	0	user_type	0	member_birth_year	8265	member_gender	8265	bike_share_for_all_trip	0	<table><tr><th></th><th>0</th></tr><tr><td>duration_sec</td><td>0.000000</td></tr><tr><td>start_time</td><td>0.000000</td></tr><tr><td>end_time</td><td>0.000000</td></tr><tr><td>start_station_id</td><td>0.107406</td></tr><tr><td>start_station_name</td><td>0.107406</td></tr><tr><td>start_station_latitude</td><td>0.000000</td></tr><tr><td>start_station_longitude</td><td>0.000000</td></tr><tr><td>end_station_id</td><td>0.107406</td></tr><tr><td>end_station_name</td><td>0.107406</td></tr><tr><td>end_station_latitude</td><td>0.000000</td></tr><tr><td>end_station_longitude</td><td>0.000000</td></tr><tr><td>bike_id</td><td>0.000000</td></tr><tr><td>user_type</td><td>0.000000</td></tr><tr><td>member_birth_year</td><td>4.506150</td></tr><tr><td>member_gender</td><td>4.506150</td></tr><tr><td>bike_share_for_all_trip</td><td>0.000000</td></tr></table> <p>dtype: float64</p>		0	duration_sec	0.000000	start_time	0.000000	end_time	0.000000	start_station_id	0.107406	start_station_name	0.107406	start_station_latitude	0.000000	start_station_longitude	0.000000	end_station_id	0.107406	end_station_name	0.107406	end_station_latitude	0.000000	end_station_longitude	0.000000	bike_id	0.000000	user_type	0.000000	member_birth_year	4.506150	member_gender	4.506150	bike_share_for_all_trip	0.000000
	0																																																																				
duration_sec	0																																																																				
start_time	0																																																																				
end_time	0																																																																				
start_station_id	197																																																																				
start_station_name	197																																																																				
start_station_latitude	0																																																																				
start_station_longitude	0																																																																				
end_station_id	197																																																																				
end_station_name	197																																																																				
end_station_latitude	0																																																																				
end_station_longitude	0																																																																				
bike_id	0																																																																				
user_type	0																																																																				
member_birth_year	8265																																																																				
member_gender	8265																																																																				
bike_share_for_all_trip	0																																																																				
	0																																																																				
duration_sec	0.000000																																																																				
start_time	0.000000																																																																				
end_time	0.000000																																																																				
start_station_id	0.107406																																																																				
start_station_name	0.107406																																																																				
start_station_latitude	0.000000																																																																				
start_station_longitude	0.000000																																																																				
end_station_id	0.107406																																																																				
end_station_name	0.107406																																																																				
end_station_latitude	0.000000																																																																				
end_station_longitude	0.000000																																																																				
bike_id	0.000000																																																																				
user_type	0.000000																																																																				
member_birth_year	4.506150																																																																				
member_gender	4.506150																																																																				
bike_share_for_all_trip	0.000000																																																																				

Gambar kiri menunjukkan hasil perhitungan jumlah missing value di setiap kolom dengan perintah `df.isnull().sum()`. Hasilnya memperlihatkan bahwa kolom seperti `start_station_id`, `start_station_name`, `end_station_id`, dan `end_station_name` masing-masing memiliki 197 data yang hilang. Sementara itu, kolom `member_birth_year` dan `member_gender` memiliki jumlah missing value yang lebih besar, yaitu 8.265. Kolom lainnya tidak memiliki missing value sama sekali karena jumlahnya tercatat nol.

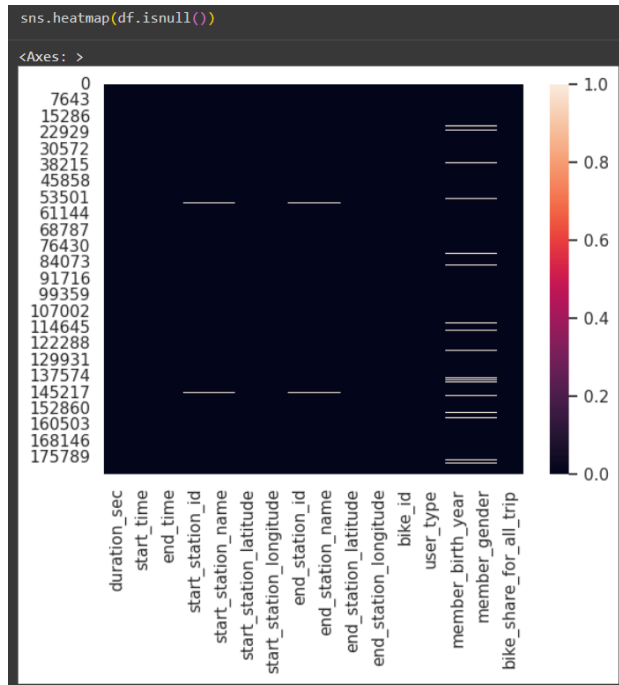
Gambar kanan memberikan informasi lanjutan mengenai persentase data yang hilang pada setiap kolom. Perhitungan ini dilakukan dengan membagi jumlah missing value pada tiap kolom dengan total baris dalam dataset, kemudian

dikalikan seratus. Dari hasil yang ditampilkan, kolom `start_station_id`, `start_station_name`, `end_station_id`, dan `end_station_name` hanya memiliki 0.107406% data yang hilang, sehingga jumlahnya tergolong sangat kecil. Sementara itu, kolom `member_birth_year` dan `member_gender` memiliki persentase yang lebih tinggi, yaitu 4.506150%. Kolom lainnya tercatat 0.000000%, artinya data di kolom tersebut lengkap tanpa ada yang kosong.

```
m = pd.concat([miss,misspercent], axis=1,keys=['Total','Missing%'])
m
```

	Total	Missing%
<code>duration_sec</code>	0	0.000000
<code>start_time</code>	0	0.000000
<code>end_time</code>	0	0.000000
<code>start_station_id</code>	197	0.107406
<code>start_station_name</code>	197	0.107406
<code>start_station_latitude</code>	0	0.000000
<code>start_station_longitude</code>	0	0.000000
<code>end_station_id</code>	197	0.107406
<code>end_station_name</code>	197	0.107406
<code>end_station_latitude</code>	0	0.000000
<code>end_station_longitude</code>	0	0.000000
<code>bike_id</code>	0	0.000000
<code>user_type</code>	0	0.000000
<code>member_birth_year</code>	8265	4.506150
<code>member_gender</code>	8265	4.506150
<code>bike_share_for_all_trip</code>	0	0.000000

Untuk menyajikan hasil penggabungan antara jumlah dan persentase missing value dalam satu tabel bisa menggunakan `pd.concat()`. Kolom "Total" memperlihatkan jumlah absolut data yang hilang, sementara kolom "Missing%" memperlihatkan persentasenya terhadap keseluruhan data. Dengan tampilan ini, lebih mudah untuk memahami seberapa besar tingkat kekosongan data pada setiap kolom, baik dari sisi jumlah maupun proporsinya.



Visualisasi missing value dilakukan dengan menggunakan heatmap dari library Seaborn. Warna yang muncul memberikan gambaran distribusi data kosong di dataset, di mana warna gelap menunjukkan data yang terisi dan warna terang menunjukkan data yang kosong. Dari heatmap terlihat adanya garis-garis terang pada kolom tertentu, yaitu pada bagian start\_station\_id, end\_station\_id, start\_station\_name, end\_station\_name, serta member\_birth\_year dan member\_gender. Hal ini menegaskan kembali hasil analisis sebelumnya bahwa kolom-kolom tersebut memang memiliki missing value.

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name	end_station_latitude	end_station_longitude	bike_id	user_type	member_birth_year	member_gender	bike_share_for_all_trip
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	True	True	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
183411	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
183412	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
183413	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
183414	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
183415	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

Gambar di atas kembali menampilkan hasil df.isnull() tetapi difokuskan pada baris tertentu untuk melihat detail data yang hilang. Dalam tampilan ini terlihat adanya nilai True pada kolom member\_birth\_year, yang menandakan bahwa pada baris tersebut memang tidak ada informasi tahun kelahiran anggota. Hal ini menjadi contoh konkret bahwa dataset memang mengandung missing value pada kolom tersebut, bukan hanya hasil perhitungan agregat semata.

```
df_copy = df_copy()
df_copy.head(10)
```

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name	end_station_latitude	end_station_longitude	bike_id	user_type	member_birth_year	member_gender	bike_share_for_all_trip
0	52185	12/10/1	01:06.0	21.0	Montgomery St BART Station (Market St at 2nd St)	37.788628	-122.400811	13.0	Commercial St at Montgomery St	37.784331	-122.402823	4602	Customer	1984.0	Male	No
1	42021	12/10/1	02:03.1	23.0	The Embarcadero at Stewart St	37.781464	-122.391034	81.0	Berry St at 4th St	37.778680	-122.393170	2535	Customer	NaH	NaH	No
2	81894	12/10/1	04:08.1	88.0	Market St at Calaveras St	37.786305	-122.426026	3.0	Power St BART Station (Market St at 4th St)	37.786378	-122.404604	8606	Customer	1972.0	Male	No
3	36480	12/10/1	02:36.0	370.0	Grove St at Masonic Ave	37.774626	-122.440546	70.0	Central Ave at Fell St	37.773211	-122.444293	9538	Subscriber	1989.0	Other	No
4	1585	12/10/1	20:44.1	7.0	Frank H Ogawa Plaza	37.804902	-122.271728	222.0	10th Ave at E 15th St	37.782714	-122.248780	4898	Subscriber	1974.0	Male	Yes
5	1763	12/10/1	19:51.0	80.0	4th St at Mission Bay Blvd S	37.775047	-122.391168	323.0	Broadway at Kalarby	37.780914	-122.405950	5200	Subscriber	1959.0	Male	No
6	1147	12/10/1	14:42.0	300.0	Palm St at Willow St	37.317298	-121.894895	312.0	San Jose Children Station	37.320732	-121.901782	3803	Subscriber	1983.0	Female	No
7	1615	12/10/1	08:02.0	10.0	Washington St at Kearny St	37.785383	-122.404770	127.0	Valencia St at 21st St	37.786768	-122.421025	6329	Subscriber	1980.0	Male	No
8	1870	12/10/1	07:08.7	16.0	Washington St at Kearny St	37.785383	-122.404770	127.0	Valencia St at 21st St	37.786768	-122.421025	6546	Subscriber	1980.0	Other	No
9	1046	12/10/1	07:17.0	16.0	Power St at Kearny St	37.780915	-122.403462	121.0	Mission Playground	37.786259	-122.421339	6481	Subscriber	1982.0	Male	No

Cuplikan isi data asli dapat dilihat dengan menggunakan perintah `df_copy.head()`. Dari cuplikan tersebut terlihat bahwa dataset berisi informasi lengkap mengenai durasi perjalanan (`duration_sec`), waktu mulai dan selesai (`start_time` dan `end_time`), lokasi awal dan akhir yang terdiri dari ID, nama, serta koordinat geografis, hingga data tambahan seperti `bike_id`, jenis pengguna (`user_type`), tahun kelahiran anggota (`member_birth_year`), jenis kelamin (`member_gender`), dan status partisipasi dalam program berbagi sepeda (`bike_share_for_all_trip`). Pada bagian ini juga tampak bahwa ada beberapa sel yang kosong, misalnya pada `member_birth_year`, sehingga memperjelas adanya data yang hilang.

```
df_copy = df_copy.dropna(how='any', subset=['start_station_id'])
df_copy = df_copy.dropna(how='any', subset=['end_station_id'])
df_copy = df_copy.dropna(how='any', subset=['start_station_name'])
df_copy = df_copy.dropna(how='any', subset=['end_station_name'])
df_copy = df_copy.dropna(how='any', subset=['member_birth_year'])
df_copy = df_copy.dropna(how='any', subset=['member_gender'])
df_copy.isnull().sum()
```

```

0
duration_sec      0
start_time        0
end_time          0
start_station_id  0
start_station_name 0
start_station_latitude 0
start_station_longitude 0
end_station_id    0
end_station_name  0
end_station_latitude 0
end_station_longitude 0
bike_id           0
user_type         0
member_birth_year 0
member_gender     0
bike_share_for_all_trip 0

dtype: int64
```

```
print(df_copy.isnull().sum())
print(f"jumlah kolom : {df_copy.shape[1]}")
print(f"jumlah baris : {df_copy.shape[0]}")
```

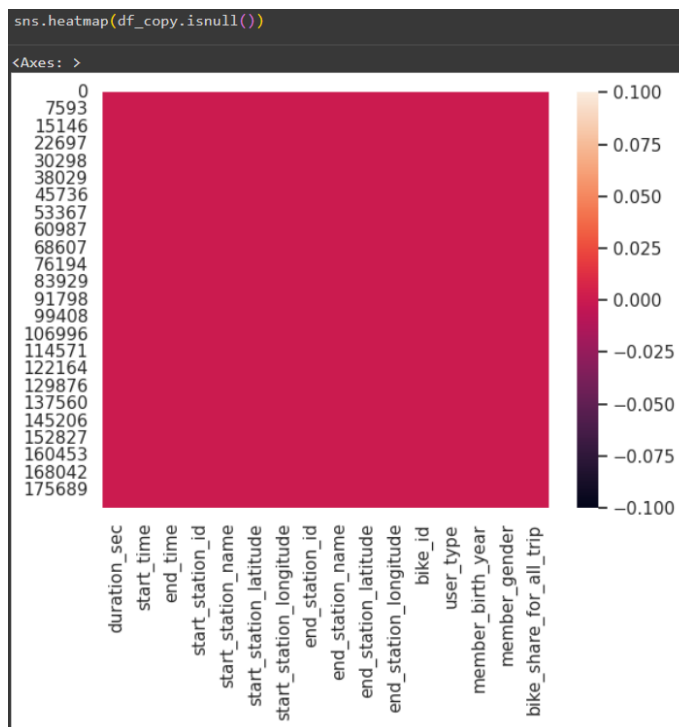
```

duration_sec      0
start_time        0
end_time          0
start_station_id  0
start_station_name 0
start_station_latitude 0
start_station_longitude 0
end_station_id    0
end_station_name  0
end_station_latitude 0
end_station_longitude 0
bike_id           0
user_type         0
member_birth_year 0
member_gender     0
bike_share_for_all_trip 0
dtype: int64
jumlah kolom : 16
jumlah baris : 174956
```

Gambar kiri menunjukkan proses pembersihan data dari missing value. Pada tahap ini digunakan perintah `dropna()` dengan parameter `subset` yang diarahkan pada kolom-kolom bermasalah, seperti `start_station_id`, `end_station_id`, `start_station_name`, `end_station_name`, `member_birth_year`, dan `member_gender`. Setelah pembersihan dilakukan, dicek kembali dengan `df_copy.isnull().sum()` dan hasilnya menunjukkan bahwa semua kolom memiliki nol missing value. Artinya, semua data kosong berhasil dihapus dari dataset, sehingga dataset yang tersisa bersih dan siap untuk digunakan dalam analisis lebih lanjut.



Gambar kanan menunjukkan proses pengecekan missing values yang dilakukan dengan menggunakan fungsi `isnull().sum()`. Dari hasil tersebut diketahui bahwa seluruh kolom yang ada pada dataset tidak memiliki nilai kosong (NaN). Jumlah kolom yang tersedia adalah 16 dan jumlah baris data sebanyak 174.956. Artinya, dataset ini cukup bersih dari sisi kelengkapan data karena tidak ada satu pun variabel yang mengalami kehilangan informasi. Hal ini merupakan kondisi yang ideal dalam analisis data, sebab jika ditemukan banyak data kosong, biasanya perlu dilakukan strategi tambahan seperti imputasi (mengganti nilai kosong dengan rata-rata/median) atau bahkan menghapus baris/kolom tertentu. Karena pada dataset ini tidak ditemukan masalah tersebut, proses pembersihan data menjadi lebih sederhana dan dapat langsung dilanjutkan ke tahap analisis berikutnya.



Hasil visualisasi heatmap dari nilai kosong dapat dilihat menggunakan fungsi `sns.heatmap(df_copy.isnull())`. Heatmap digunakan untuk memberikan gambaran visual mengenai distribusi nilai kosong di dalam dataset. Warna yang muncul pada heatmap adalah warna solid yang seragam, tanpa ada celah atau blok warna lain, yang berarti seluruh kolom dan baris tidak memiliki nilai kosong. Dengan kata lain, hasil visualisasi ini konsisten dengan temuan pada gambar pertama bahwa dataset tidak memiliki missing values. Visualisasi seperti ini penting digunakan sebagai validasi, karena kadang hasil numerik saja kurang meyakinkan tanpa dukungan visual.

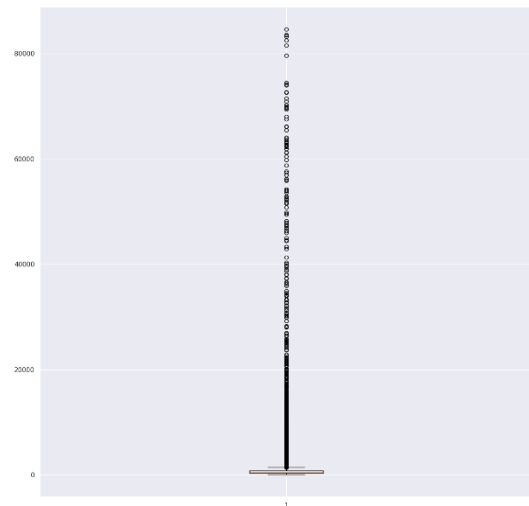
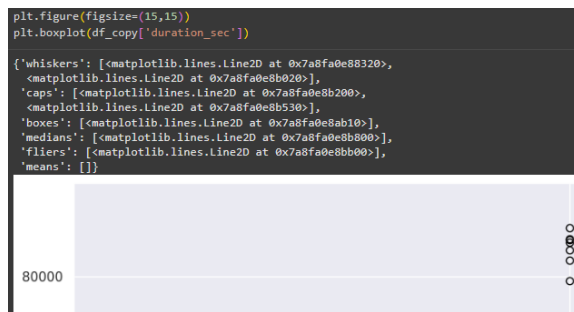
## - Membuat copy dataset

```
df_copy.describe(include="all")
```

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name	end_station_latitude	end_station_longitude	bike_id	user_type	member_birth_year	member_gender	bike_share_for_all_trip
count	174956.000000	174956	174956	174956.000000	174956	174956.000000	174956.000000	174956.000000	174956	174956.000000	174956.000000	174956	174956.000000	174956	174956	174956
unique	NaN	35732	35737	NaN	329	NaN	NaN	NaN	329	NaN	NaN	NaN	NaN	2	NaN	3
top	NaN	12/19/2	22:34:4	NaN	Market St at 18th St	NaN	NaN	NaN	San Francisco Caltrans Station 2 (Townsend St	NaN	NaN	NaN	Subscriber	NaN	Male	Na
freq	NaN	12	19	NaN	3609	NaN	NaN	NaN	4624	NaN	NaN	NaN	152390	NaN	130204	157809
mean	783.391556	NaN	NaN	139.862952	NaN	37.771218	-122.351758	136.064769	NaN	37.771412	-122.351323	4482.570441	NaN	1984.803262	NaN	NaN
std	1642.187831	NaN	NaN	111.642980	NaN	6.198395	8.117736	111.335413	NaN	6.198396	8.117738	1459.140968	NaN	19.110831	NaN	NaN
min	61.000000	NaN	NaN	3.000000	NaN	37.357298	-122.453795	3.000000	NaN	37.357298	-122.453795	11.000000	NaN	1878.000000	NaN	NaN
25%	323.000000	NaN	NaN	47.000000	NaN	37.779487	-122.411981	44.000000	NaN	37.779487	-122.411647	3789.000000	NaN	1980.000000	NaN	NaN
50%	519.000000	NaN	NaN	184.000000	NaN	37.786768	-122.386279	181.000000	NaN	37.781818	-122.387437	4960.000000	NaN	1987.000000	NaN	NaN
75%	789.000000	NaN	NaN	238.000000	NaN	37.787329	-122.383993	238.000000	NaN	37.787473	-122.386533	5595.000000	NaN	1992.000000	NaN	NaN
max	84546.000000	NaN	NaN	385.000000	NaN	37.888222	-121.874119	385.000000	NaN	37.888222	-121.874119	6645.000000	NaN	2001.000000	NaN	NaN

Dengan menuliskan `describe(include="all")` maka statistik deskriptif dari seluruh kolom dalam dataset, baik numerik maupun kategorikal akan ditampilkan. Untuk variabel numerik, ditampilkan informasi seperti jumlah data (count), nilai rata-rata (mean), standar deviasi (std), nilai minimum, kuartil (25%, 50%, 75%), hingga nilai maksimum. Sementara untuk variabel kategorikal, ditampilkan jumlah kategori unik (unique), nilai yang paling sering muncul (top), serta frekuensi kemunculannya (freq). Dari sini dapat dilihat, misalnya, distribusi `member_birth_year`, sebaran ID sepeda (`bike_id`), serta nama stasiun asal maupun tujuan yang paling sering digunakan. Statistik deskriptif seperti ini memberikan gambaran awal mengenai struktur data dan menjadi dasar untuk mendeteksi ketidakwajaran, misalnya nilai tahun lahir yang terlalu kecil atau terlalu besar, atau adanya stasiun dengan nama yang tidak lengkap.

## - Cek Outlier



Tampilkan boxplot kolom untuk kolom `duration_sec`, yaitu variabel yang merepresentasikan lama perjalanan dalam hitungan detik. Dari boxplot ini terlihat adanya banyak titik-titik kecil di luar batas whisker, yang merupakan indikasi outlier. Outlier ini bisa berupa perjalanan yang sangat singkat atau sangat lama dibandingkan mayoritas data lainnya. Keberadaan outlier dapat memengaruhi analisis statistik, misalnya membuat rata-rata durasi perjalanan menjadi lebih tinggi atau rendah dari kondisi sebenarnya. Oleh karena itu, identifikasi outlier merupakan langkah penting dalam proses data cleaning, sebelum dilakukan analisis lebih lanjut.

```

Q1 = df_copy['duration_sec'].quantile(0.25)
Q3 = df_copy['duration_sec'].quantile(0.75)
IQR = Q3 - Q1
outlierQ3 = Q3 + 1.5 * IQR
df_copy = df_copy[df_copy['duration_sec'] < outlierQ3]
df_copy['duration_sec'].describe()

duration_sec
count    165610.000000
mean         550.095840
std         304.148098
min           61.000000
25%         314.000000
50%         488.000000
75%         731.000000
max        1487.000000

dtype: float64

print(f"jumlah kolom : {df_copy.shape[1]}")
print(f"jumlah baris : {df_copy.shape[0]}")

jumlah kolom : 16
jumlah baris : 165610

```

Proses pembersihan data untuk kolom `member_birth_year` menggunakan metode IQR (Interquartile Range). IQR adalah salah satu teknik yang sering digunakan untuk mendeteksi outlier pada data numerik. Caranya adalah dengan menghitung selisih antara kuartil ketiga (Q3) dan kuartil pertama (Q1), lalu menentukan batas bawah ( $Q1 - 1.5IQR$ ) dan batas atas ( $Q3 + 1.5IQR$ ). Data yang berada di luar batas ini dianggap sebagai outlier. Setelah proses ini dilakukan, jumlah baris dataset berkurang dari 174.956 menjadi 159.433. Artinya, terdapat sejumlah data tahun lahir yang dianggap tidak masuk akal (misalnya terlalu tua atau terlalu muda untuk menjadi pengguna sepeda) yang berhasil dihapus. Dengan demikian, distribusi data tahun lahir menjadi lebih masuk akal dan lebih representatif.



Tampilkan boxplot kolom untuk kolom `member_birth_year`, yaitu variabel yang merepresentasikan tahun kelahiran pengguna. Dari boxplot ini terlihat adanya banyak titik-titik kecil di luar batas whisker, yang merupakan indikasi outlier. Outlier ini terlihat berada pada Q1 atau di bawah dari batas whisker.

```

Q1 = df_copy["member_birth_year"].quantile(0.25)
Q3 = df_copy["member_birth_year"].quantile(0.75)
IQR = Q3 - Q1
outlierQ1 = Q1 - 1.5 * IQR
df_copy = df_copy[(df_copy["member_birth_year"] > outlierQ1)]
df_copy["member_birth_year"].describe()

member_birth_year
count    159433.000000
mean      1985.948279
std         8.300881
min       1963.000000
25%       1981.000000
50%       1988.000000
75%       1992.000000
max       2001.000000
dtype: float64

print(f"jumlah kolom : {df_copy.shape[1]}")
print(f"jumlah baris : {df_copy.shape[0]}")

jumlah kolom : 16
jumlah baris : 159433

```

Pembersihan data untuk kolom `member_birth_year` menggunakan metode IQR (Interquartile Range). Caranya adalah dengan menghitung selisih antara kuartil ketiga (Q3) dan kuartil pertama (Q1), lalu menentukan batas bawah ( $Q1 - 1.5IQR$ ) dan batas atas ( $Q3 + 1.5IQR$ ). Data yang berada di luar batas ini dianggap sebagai outlier. Setelah proses ini dilakukan, jumlah baris dataset berkurang dari 174.956 menjadi 159.433. Artinya, terdapat sejumlah data tahun lahir yang dianggap tidak masuk akal (misalnya terlalu tua atau terlalu muda untuk menjadi pengguna sepeda) yang berhasil dihapus. Dengan demikian, distribusi data tahun lahir menjadi lebih masuk akal dan lebih representatif.

- Standardisasi format kolom (huruf kecil, strip whitespace)

```

df.columns = df.columns.str.strip().str.lower().str.replace(" ", "_")
df.columns

Index(['duration_sec', 'start_time', 'end_time', 'start_station_id',
      'start_station_name', 'start_station_latitude',
      'start_station_longitude', 'end_station_id', 'end_station_name',
      'end_station_latitude', 'end_station_longitude', 'bike_id', 'user_type',
      'member_birth_year', 'member_gender', 'bike_share_for_all_trip'],
      dtype='object')

```

Potongan kode `df.columns = df.columns.str.strip().str.lower().str.replace(" ", "_")` digunakan untuk melakukan proses pembersihan dan standarisasi nama kolom pada dataframe. Pertama, fungsi `str.strip()` berperan untuk menghapus spasi berlebih yang mungkin ada di awal maupun akhir nama kolom sehingga tidak menimbulkan error saat pemanggilan. Kedua, `str.lower()` mengubah semua huruf dalam nama kolom menjadi huruf kecil agar lebih konsisten, karena Python bersifat case-sensitive dan bisa membedakan huruf besar serta huruf kecil. Terakhir, `str.replace(" ", "_")` digunakan untuk mengganti spasi yang terdapat di antara kata dengan garis bawah (underscore), sehingga nama kolom lebih mudah dipanggil dan sesuai dengan aturan penamaan variabel di Python.

## - Deteksi dan penghapusan data duplikat

```
df_copy.describe(include="all")
```

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name	end_station_latitude	end_station_longitude	bike_id	user_type	member_birth_year	member_gender	bike_share_for_all_trip
count	159433.000000	159433	159433	159433.000000	159433	159433.000000	159433.000000	159433.000000	159433	159433.000000	159433.000000	159433	159433	159433.000000	159433	159433
unique	NaN	35596	35573	NaN	329	NaN	NaN	NaN	329	NaN	NaN	NaN	2	NaN	3	2
top	NaN	55.66.9	22.24.4	NaN	Market St at 10th St	NaN	NaN	NaN	San Francisco Caltrans Station 2 (Townsend St	NaN	NaN	NaN	Subscriber	NaN	Male	No
freq	NaN	15	16	NaN	3478	NaN	NaN	NaN	4474	NaN	NaN	NaN	146135	NaN	119051	144140
mean	548.899795	NaN	NaN	136.945908	NaN	37.778419	-122.350634	135.954238	NaN	37.778594	-122.350102	4487.188232	NaN	1985.948279	NaN	NaN
std	302.984342	NaN	NaN	118.888220	NaN	8.182105	8.118234	110.181727	NaN	8.182066	8.118735	1654.570987	NaN	8.308881	NaN	NaN
min	61.000000	NaN	NaN	3.000000	NaN	37.317286	-122.453795	3.000000	NaN	37.317288	-122.453795	11.000000	NaN	1983.000000	NaN	NaN
25%	313.000000	NaN	NaN	48.000000	NaN	37.778883	-122.411738	44.000000	NaN	37.778487	-122.411306	3824.000000	NaN	1981.000000	NaN	NaN
50%	487.000000	NaN	NaN	194.000000	NaN	37.788626	-122.397437	100.000000	NaN	37.788955	-122.397886	4969.000000	NaN	1988.000000	NaN	NaN
75%	729.000000	NaN	NaN	239.000000	NaN	37.787329	-122.280192	233.000000	NaN	37.787329	-122.283993	5585.000000	NaN	1982.000000	NaN	NaN
max	1487.000000	NaN	NaN	388.000000	NaN	37.888222	-121.874119	388.000000	NaN	37.888222	-121.874119	6645.000000	NaN	2001.000000	NaN	NaN

Setelah proses pembersihan outlier dilakukan, deteksi data dupliat menggunakan `describe(include="all")`, dari ringkasan statistik terbaru terlihat bahwa distribusi data pada kolom `member_birth_year` menjadi lebih wajar, dengan nilai minimum dan maksimum yang masuk akal untuk usia pengguna sepeda. Hal ini memastikan bahwa data yang digunakan dalam analisis selanjutnya lebih berkualitas, karena sudah terbebas dari outlier ekstrem yang bisa menimbulkan bias. Statistik deskriptif setelah pembersihan ini juga menunjukkan konsistensi data pada kolom lain yang tidak mengalami perubahan signifikan.

```
df_copy.duplicated()
```

0	
6	False
9	False
10	False
11	False
12	False
...	...
183411	False
183412	True
183413	True
183414	True
183415	True

159433 rows x 1 columns

dtype: bool

```
df_copy = df_copy.drop_duplicates()
print(f"jumlah kolom : {df_copy.shape[1]}")
print(f"jumlah baris : {df_copy.shape[0]}")
```

jumlah kolom : 16  
jumlah baris : 159429

Pengecekan data duplikat menggunakan `df_copy.duplicated()`. Dari output terlihat bahwa terdapat beberapa baris data yang bernilai `True`, artinya data tersebut terdeteksi sebagai duplikat. Keberadaan duplikasi bisa menimbulkan masalah dalam analisis karena dapat menghitung satu peristiwa lebih dari sekali. Setelah dilakukan proses penghapusan dengan `drop_duplicates()`, jumlah baris dataset berkurang dari 159.433 menjadi 159.429. Artinya, terdapat 4 baris duplikat yang berhasil dihapus. Dengan demikian, dataset kini bebas dari duplikasi dan lebih siap untuk dianalisis secara akurat.

## - Format tanggal dan konversi tipe data

```

date_columns = [col for col in df.columns if 'date' in col or 'time' in col]
for col in date_columns:
    df[col] = pd.to_datetime(df[col], errors='coerce')
print("Kolom yang dikonversi ke datetime:", date_columns)
df.head()

```

	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name	end_station_latitude	end_station_longitude	bike_id	user_type	member_birth_year	member_gender	bike_share_for_all_trip
0	NaT	2025-06-26 01:55:00	21.0	Montgomery St BART Station (Market St at 4th St)	37.780625	-122.400811	13.0	Commercial St at Montgomery St	37.794231	-122.402623	4002	Customer	1984.0	Male	No
1	NaT	NaT	23.0	The Embarcadero at Duval St	37.791484	-122.391034	81.0	Berry St at 4th St	37.770880	-122.393170	2538	Customer	NaT	NaT	No
2	2025-06-26 10:15:12	NaT	88.0	Market St at Duval St	37.780305	-122.408026	3.0	Powell St BART Station (Market St at 4th St)	37.788375	-122.409044	5005	Customer	1972.0	Male	No
3	NaT	2025-06-26 12:38:48	378.0	Glenn St at Mission Ave	37.774038	-122.448348	70.0	Central Ave at F st	37.773311	-122.444263	9038	Subscriber	1988.0	Other	No
4	1988	NaT	2025-06-26 20:44:08	Franklin Square Plaza	37.804582	-122.271738	222.0	10th Ave at E 10th St	37.782714	-122.248760	4988	Subscriber	1974.0	Male	Yes

Potongan kode tersebut digunakan untuk mendeteksi dan mengonversi kolom yang berhubungan dengan tanggal atau waktu pada dataframe menjadi format datetime. Pertama, baris `date_columns = [col for col in df.columns if 'date' in col or 'time' in col]` membuat sebuah list berisi nama kolom yang mengandung kata 'date' atau 'time'. Hal ini dilakukan agar kita tidak perlu menentukan kolom tanggal atau waktu secara manual, melainkan bisa menemukannya secara otomatis dari nama kolom. Selanjutnya, dilakukan perulangan `for col in date_columns`: yang akan mengambil setiap kolom dalam list tersebut dan mengonversinya ke tipe data datetime menggunakan fungsi `pd.to_datetime(df[col], errors='coerce')`. Argumen `errors='coerce'` memastikan bahwa jika ada nilai yang tidak bisa dikonversi ke format tanggal, maka akan otomatis diubah menjadi NaT (Not a Time) agar tidak menimbulkan error.

Setelah proses konversi selesai, perintah `print("Kolom yang dikonversi ke datetime:", date_columns)` akan menampilkan daftar kolom yang berhasil diubah ke format datetime. Dengan langkah ini, tipe data kolom yang berkaitan dengan waktu tidak lagi berupa string, melainkan sudah sesuai dengan format tanggal dan waktu yang dapat dikenali Pandas. Dalam hal ini, kolom yang telah dikonversi menjadi datetime ialah kolom `start_time` dan `end_time`

## 4. ANALISIS DATA

### 4.1 Eksplorasi Variabel Utama

Pilih variable yang akan digunakan dalam grafik :

- User\_type
- Member\_gender
- Duration\_sec
- Start\_station\_name
- Member\_birth\_year
- Start\_station\_latitude
- End\_station\_latitude
- Start\_station\_longitude
- End\_station\_longitude

## 4.2 Visualisasi Data

- Memastikan tidak ada data duplikat

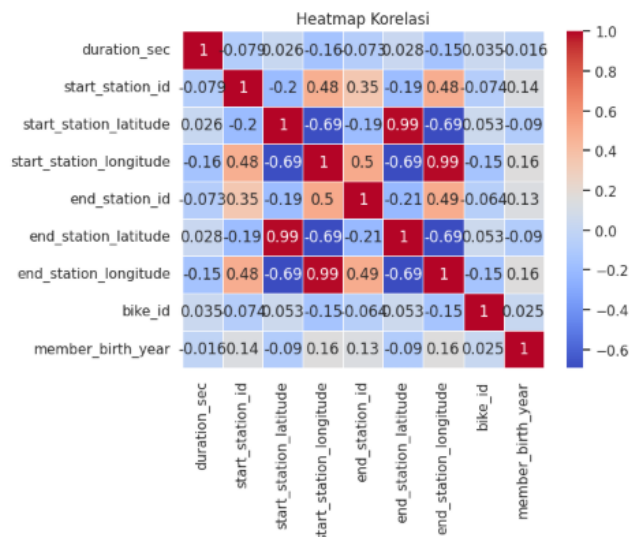
```
df_copy.duplicated().sum()
np.int64(0)
```

Terlihat bahwa dalam data sudah tidak terdapat duplikasi karena pada tahap sebelumnya telah dilakukan pembersihan data.

- Menampilkan Korelasi antar kolom untuk analisis data

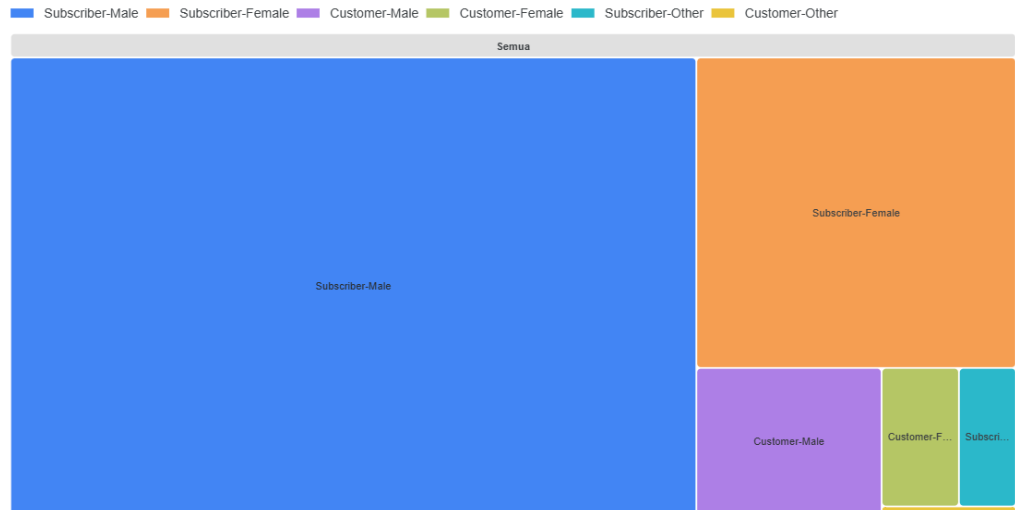
```
df_copy_corr = df_copy[["duration_sec", "start_station_id",
                        "start_station_latitude", "start_station_longitude",
                        "end_station_id", "end_station_latitude",
                        "end_station_longitude", "bike_id", "member_birth_year"]]
corr_matrix = df_copy_corr.corr()

sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Heatmap Korelasi")
plt.show()
```



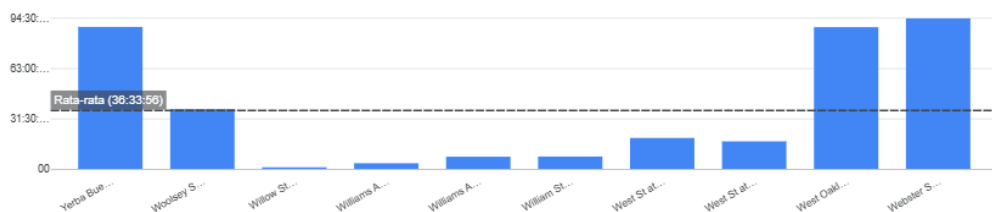
Gambar tersebut menampilkan sebuah heatmap korelasi yang menggambarkan hubungan antar variabel dalam dataset terkait penggunaan sepeda. Variabel yang dianalisis antara lain durasi penggunaan sepeda (`duration_sec`), identitas stasiun awal dan akhir (`start_station_id`, `end_station_id`), koordinat geografis stasiun (latitude dan longitude), identitas sepeda (`bike_id`), serta tahun kelahiran anggota (`member_birth_year`). Warna pada heatmap menunjukkan tingkat korelasi, dengan skala dari -1 hingga 1. Korelasi positif yang tinggi ditunjukkan dengan warna merah (mendekati 1), sedangkan korelasi negatif ditunjukkan dengan warna biru (mendekati -1). Misalnya, terlihat bahwa terdapat korelasi sangat kuat antara koordinat stasiun awal dan stasiun akhir baik pada latitude maupun longitude, yang ditunjukkan dengan nilai di atas 0.9, menandakan bahwa lokasi awal dan akhir perjalanan cenderung berdekatan. Sebaliknya, variabel seperti `duration_sec` dan `member_birth_year` tidak memiliki korelasi kuat dengan variabel lain, karena nilai korelasinya mendekati nol.

- Menampilkan hasil visualisasi analisis dengan chart yang tepat
  - Treemap - Bagaimana perbandingan frekuensi penyewaan sepeda antara customer dan subscriber?

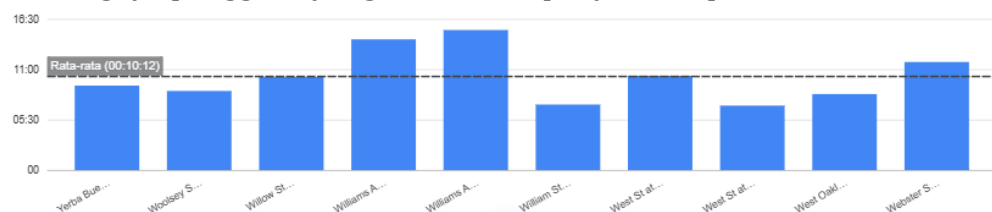


Mayoritas pengguna adalah Subscriber-Male dengan persentase terbesar, yaitu 96.217 (68,2%), disusul oleh Subscriber-Female sebesar 30.448 (21,6%). Sementara itu, kelompok Customer-Male memiliki porsi 8.342 (5,9%), sedangkan kategori lain seperti Customer-Female, Subscriber-Other, dan Customer-Other hanya mencakup persentase yang sangat kecil. Hal ini menggambarkan bahwa pengguna layanan didominasi oleh laki-laki, khususnya yang berstatus pelanggan tetap (subscriber).

- Bar chart - Apa stasiun yang sering didatangi pengguna dan memiliki peluang durasi penyewaan lebih besar?



Stasiun yang masih memiliki sedikit penyewa ialah Willow st, total durasi yang tercatat hanya sebesar 1 jam, sangat kontras dengan rata-rata penyewaan yang mampu mencetak 30jam. Hal ini menandakan bahwa kurangnya pengguna yang melakukan penyewaan pada Willow st.

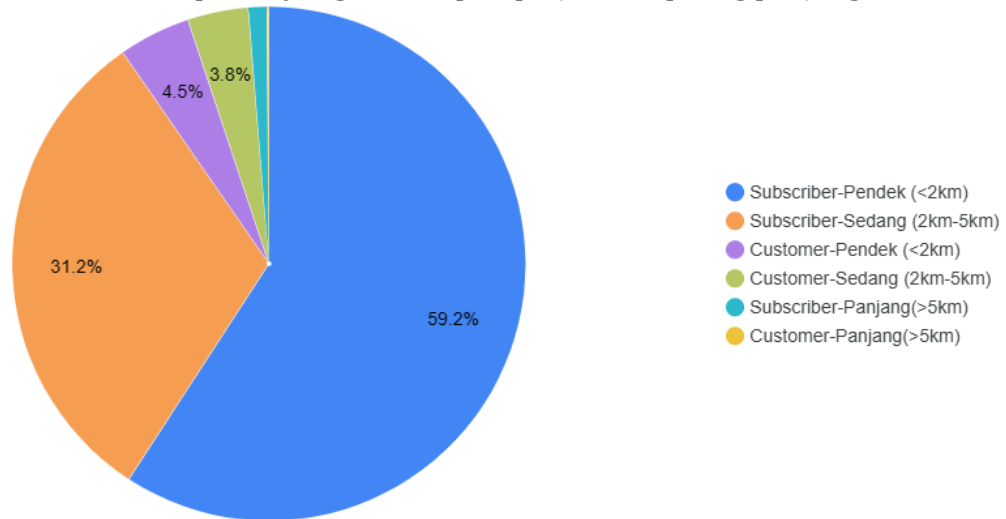


Meski kurangnya pengguna yang melakukan penyewaan pada Willow st, namun rata-rata durasi penggunaan pada stasiun ini cukup tinggi, yaitu



rata-rata 10 menit. Hal ini menandakan bahwa meski jarang penyewaan terjadi di stasiun tersebut, namun penyewa cukup menghabiskan waktu lebih panjang.

- Pie chart – Siapakah yang menempuh perjalanan paling panjang?



Mayoritas perjalanan dilakukan oleh Subscriber dengan jarak pendek (<2 km) yang mencapai 59,2%, diikuti oleh Subscriber dengan jarak sedang (2–5 km) sebesar 31,2%. Sementara itu, porsi perjalanan dari pelanggan kategori Customer relatif kecil, seperti Customer-Pendek (<2 km) sebesar 4,5% dan Customer-Sedang (2–5 km) sebesar 3,8%. Adapun perjalanan jarak panjang (>5 km), baik oleh subscriber maupun customer, hanya menyumbang persentase sangat kecil.

## 5. TEMUAN & INSIGHT

- Treemap - Bagaimana perbandingan frekuensi penyewaan sepeda antara customer dan subscriber?

Berdasarkan diagram distribusi pengguna, dapat ditemukan bahwa mayoritas pengguna layanan berasal dari kelompok Subscriber-Male dengan persentase yang sangat dominan yaitu 68,2%, disusul oleh Customer-Male sebesar 21,6%. Hal ini menunjukkan bahwa laki-laki menjadi segmen utama pengguna layanan, khususnya mereka yang berstatus pelanggan tetap. Sementara itu, proporsi pengguna perempuan masih rendah, baik dari sisi subscriber maupun customer, ditambah kategori lain yang hampir tidak signifikan. Temuan ini mengindikasikan bahwa meskipun layanan sudah berhasil menjangkau pasar pria dengan baik, terdapat peluang besar untuk memperluas pasar melalui peningkatan keterlibatan perempuan dan konversi customer menjadi subscriber.

- Bar Chart - Apa stasiun yang sering didatangi pengguna dan memiliki peluang durasi penyewaan lebih besar?

Berdasarkan kedua grafik sebelumnya, terlihat adanya perbedaan pola penggunaan sepeda di tiap stasiun. Pada grafik pertama, rata-rata durasi penyewaan sepeda tercatat sekitar 36 jam 33 menit. Beberapa stasiun seperti Yerba Buena, West Oakland, dan Webster St. menonjol dengan durasi penyewaan yang jauh di atas rata-rata, yang mengindikasikan bahwa sepeda di stasiun ini lebih sering digunakan untuk perjalanan panjang. Sebaliknya, stasiun seperti Willow St. dan Williams Ave. menunjukkan durasi penyewaan yang sangat rendah, sehingga penggunaannya cenderung untuk perjalanan singkat.

Sementara itu, pada grafik kedua, rata-rata lama perjalanan per trip hanya sekitar 10 menit 12 detik. Meskipun total durasi penyewaan di beberapa stasiun sangat tinggi, lama perjalanan rata-rata per trip di hampir semua stasiun relatif singkat dan tidak jauh berbeda. Stasiun Williams Ave. menjadi pengecualian dengan rata-rata durasi perjalanan tertinggi mendekati 15 menit. Hal ini mengindikasikan adanya kemungkinan pola peminjaman berulang atau adanya pengguna dengan durasi sewa sangat lama yang memengaruhi total rata-rata.

Temuan ini menunjukkan bahwa terdapat perbedaan karakteristik antara stasiun yang cenderung digunakan untuk perjalanan panjang dengan stasiun yang hanya melayani perjalanan singkat harian. Perbedaan pola ini bisa dijadikan dasar strategi, misalnya mendorong minat pada stasiun dengan durasi rendah melalui promosi atau peningkatan fasilitas, serta memperkuat layanan di stasiun dengan durasi tinggi agar lebih optimal mendukung perjalanan jarak jauh.

- Pie Chart - Siapakah yang menempuh perjalanan paling panjang?

Berdasarkan distribusi perjalanan, terlihat bahwa mayoritas pengguna adalah Subscriber dengan perjalanan pendek (<2 km) yang mencapai 59,2%, diikuti oleh Subscriber dengan perjalanan sedang (2–5 km) sebesar 31,2%. Sementara itu,

kontribusi perjalanan dari Customer jauh lebih kecil, baik untuk kategori pendek maupun sedang, serta hampir tidak ada pada kategori jarak panjang (>5 km). Hal ini menunjukkan bahwa layanan didominasi oleh pengguna subscriber dengan preferensi perjalanan jarak dekat. Dengan kata lain, terdapat pola bahwa pelanggan berlangganan lebih nyaman memanfaatkan sepeda untuk kebutuhan jarak pendek sehari-hari, sementara potensi customer dan perjalanan jarak jauh masih rendah.

## 6. PELUANG BISNIS

- Treemap - Bagaimana perbandingan frekuensi penyewaan sepeda antara customer dan subscriber?
  - a. Program Konversi Customer → Subscriber
  - b. Segmentasi Layanan Berdasarkan Gender & Preferensi
- Bar Chart - Apa stasiun yang sering didatangi pengguna dan memiliki peluang durasi penyewaan lebih besar?
  - a. Optimalisasi Stasiun Sepi melalui Peningkatan Daya Tarik
  - b. Strategi Promosi Berbasis Lokasi
  - c. Program Peningkatan Kualitas Armada Sepeda
  - d. Gamifikasi & Program Loyalitas Pelanggan
- Pie Chart - Siapakah yang menempuh perjalanan paling panjang?
  - a. Kerja Sama dengan Bisnis Lokal di Sekitar Stasiun
  - b. Program Green Loyalty (Poin Hijau)

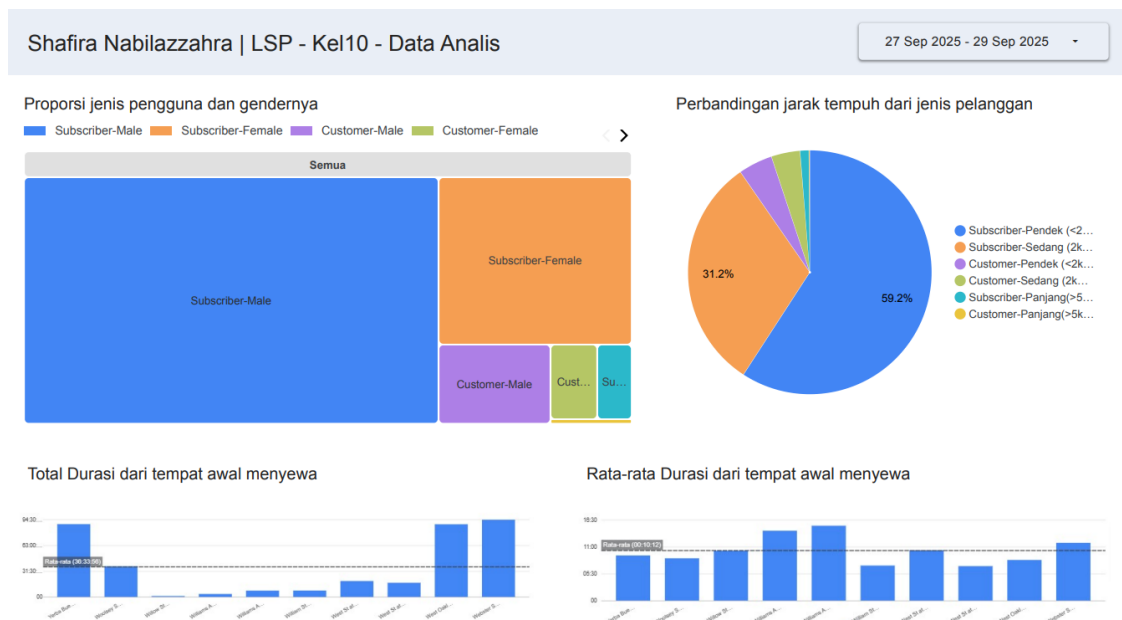
## 7. KESIMPULAN DAN REKOMENDASI

- Treemap – Bagaimana perbandingan frekuensi penyewaan sepeda antara customer dan subscriber?
  - a. Buat promosi khusus untuk customer (terutama laki-laki) agar berlangganan, misalnya paket hemat atau bonus menit perjalanan tambahan jika upgrade ke subscriber.
  - b. Tawarkan fitur personalisasi dalam aplikasi, seperti rekomendasi rute aman bagi perempuan atau fitur sosial yang bisa menarik minat kelompok pengguna yang masih kecil.
- Bar Chart - Apa stasiun yang sering didatangi pengguna dan memiliki peluang durasi penyewaan lebih besar?
  - a. Menambahkan spot foto aesthetic pada stasiun yang jarang dikunjungi

- b. Tambahkan fitur kupon/voucher diskon otomatis pada stasiun yang jarang dikunjungi
  - c. Perbaiki/percantik sepeda di stasiun yang memiliki rata-rata penyewaan rendah
  - d. Buat fitur aplikasi baru, setelah melakukan penyewaan sepeda selama lebih dari 10 menit maka akan mendapatkan coin. Coin bisa dipakai untuk membeli kupon/voucher diskon, semakin sering pengguna menyewa sepeda, maka coin semakin bertambah, memungkinkan loyalitas customer akan berubah menjadi subscriber.
- Pie Chart - Siapakah yang menempuh perjalanan paling panjang?
- a. Karena mayoritas perjalanan subscriber bersifat jarak pendek, bisa dibuat kolaborasi dengan kafe, minimarket, atau pusat perbelanjaan sekitar stasiun untuk menawarkan diskon khusus bagi pengguna sepeda. Hal ini mendorong aktivitas ekonomi lokal sekaligus meningkatkan loyalitas pengguna.
  - b. Setiap kilometer perjalanan bisa dikonversi menjadi poin ramah lingkungan yang bisa ditukar dengan hadiah. Misalnya 3 km = 1 pohon ditanam, sehingga meningkatkan branding eco-friendly.

## LAMPIRAN-LAMPIRAN

### - Visualisasi tambahan (Dashboard)



## - Tabel statistik deskriptif

#f_numpy.describe(include='all')																
	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name	end_station_latitude	end_station_longitude	bike_id	user_type	member_birth_year	member_gender	bike_share_for_all_trip
count	159429.000000	94287	81910	159429.000000	159429	159429.000000	159429.000000	159429.000000	159429	159429.000000	159429.000000	159429	159429.000000	159429	159429	159429
unique	NaN	NaN	NaN	NaN	329	NaN	NaN	NaN	329	NaN	NaN	NaN	2	NaN	3	2
top	NaN	NaN	NaN	NaN	Market St at 10th St	NaN	NaN	NaN	San Francisco Caltrans Station 2 (Townsend St.	NaN	NaN	NaN	Subscriber	NaN	Male	NaN
freq	NaN	NaN	NaN	NaN	2479	NaN	NaN	NaN	4474	NaN	NaN	NaN	149121	NaN	119247	144127
mean	548.818145	2025-09-29 11:56:26.902480Z	2025-09-29 12:03:16.3905980Z	138.845078	NaN	37.775421	-122.350637	125.853911	NaN	37.775957	-122.350104	4487.199129	NaN	1985.946234	NaN	NaN
min	61.000000	2025-09-29 00:00:00	2025-09-29 00:00:00	3.000000	NaN	37.317286	-122.453709	3.000000	NaN	37.317286	-122.453705	11.000000	NaN	1983.000000	NaN	NaN
25%	313.000000	2025-09-29 08:07:08	2025-09-29 08:00:24	49.000000	NaN	37.775983	-122.411758	49.000000	NaN	37.775937	-122.411306	3524.000000	NaN	1981.000000	NaN	NaN
50%	487.000000	2025-09-29 11:48:24	2025-09-29 12:06:45	104.000000	NaN	37.780525	-122.387437	105.000000	NaN	37.780955	-122.387088	4950.000000	NaN	1985.000000	NaN	NaN
75%	726.000000	2025-09-29 17:48:04	2025-09-29 18:00:46	239.000000	NaN	37.797509	-122.380182	239.000000	NaN	37.797536	-122.380903	5596.000000	NaN	1982.000000	NaN	NaN
max	1487.000000	2025-09-29 23:59:54	2025-09-29 23:59:54	385.000000	NaN	37.396232	-121.874119	385.000000	NaN	37.396232	-121.874119	6548.000000	NaN	2001.000000	NaN	NaN
std	302.893348	NaN	NaN	110.838108	NaN	9.102161	9.118232	105.183942	NaN	9.103992	9.118731	1884.567483	NaN	8.303913	NaN	NaN

## - Potongan kode Python dari notebook

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import pyplot as plt
sns.set()
%matplotlib inline
```

```
df = pd.read_csv('fordgobike.csv')
df.head()
```

```
df.shape
```

```
df.size
```

```
df.columns
```

```
df.dtypes
```

```
df.info()
```

```
df.describe(include='all')
```

```
df.nunique()
```

```
df.isnull()
```

```
miss = df.isnull().sum()
miss
```

```
misspercent = (df.isnull().sum()/len(df))*100
misspercent
```

```

m = pd.concat([miss, misspercent], axis=1, keys=['Total','Missing%'])
m

sns.heatmap(df.isnull())

df.isnull()

df_copy = df.copy()
df_copy.head(10)

df_copy = df_copy.dropna(how='any', subset=['start_station_id'])
df_copy = df_copy.dropna(how='any', subset=['end_station_id'])
df_copy = df_copy.dropna(how='any', subset=['start_station_name'])
df_copy = df_copy.dropna(how='any', subset=['end_station_name'])
df_copy = df_copy.dropna(how='any', subset=['member_birth_year'])
df_copy = df_copy.dropna(how='any', subset=['member_gender'])
df_copy.isnull().sum()

print(df_copy.isnull().sum())
print(f'jumlah kolom : {df_copy.shape[1]}')
print(f'jumlah baris : {df_copy.shape[0]}')

sns.heatmap(df_copy.isnull())

df_copy.describe(include='all')

plt.figure(figsize=(15,15))
plt.boxplot(df_copy['duration_sec'])

Q1 = df_copy['duration_sec'].quantile(0.25)
Q3 = df_copy['duration_sec'].quantile(0.75)
IQR = Q3 - Q1
outlierQ3 = Q3 + 1.5 * IQR
df_copy = df_copy[df_copy['duration_sec'] < outlierQ3]
df_copy['duration_sec'].describe()

print(f'jumlah kolom : {df_copy.shape[1]}')
print(f'jumlah baris : {df_copy.shape[0]}')

```

```

plt.figure(figsize=(15,15))
plt.boxplot(df_copy['member_birth_year'])

Q1 = df_copy["member_birth_year"].quantile(0.25)
Q3 = df_copy["member_birth_year"].quantile(0.75)
IQR = Q3 - Q1
outlierQ1 = Q1 - 1.5 * IQR
df_copy = df_copy[(df_copy["member_birth_year"] > outlierQ1)]
df_copy["member_birth_year"].describe()

print(f'jumlah kolom : {df_copy.shape[1]}')
print(f'jumlah baris : {df_copy.shape[0]}')

df_copy.describe(include='all')

df.columns = df.columns.str.strip().str.lower().str.replace(" ", "_")
df.columns

df_copy.duplicated()

df_copy = df_copy.drop_duplicates()
print(f'jumlah kolom : {df_copy.shape[1]}')
print(f'jumlah baris : {df_copy.shape[0]}')

date_columns = [col for col in df_copy.columns if 'date' in col or 'time' in col]
for col in date_columns:
    df_copy[col] = pd.to_datetime(df[col], errors='coerce')
print("Kolom yang dikonversi ke datetime:", date_columns)
df_copy.dtypes
df_copy.head()

df_copy.duplicated().sum()

df_copy_corr = df_copy[["duration_sec", "start_station_id",
                        "start_station_latitude", "start_station_longitude",
                        "end_station_id", "end_station_latitude",
                        "end_station_longitude", "bike_id", "member_birth_year"]]
corr_matrix = df_copy_corr.corr()

sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", linewidths=0.5)

```

```
plt.title("Heatmap Korelasi")  
plt.show()  
  
df_copy.describe(include='all')
```