

ANALISIS DATASET *TOP 100 STEAM GAMES NOVEMBER 2021* MENGUNAKAN BAHASA PEMROGRAMAN PYTHON

Laporan Tugas Besar

**Disusun sebagai syarat tugas besar
Pengenalannya Komputasi
KU1102-72**

Oleh:

- 1. MATTHEW MAHENDRA. NIM: 16521497**
- 2. HENRY ANAND SEPTIAN RADITYO. NIM: 16521507**
- 3. RICHARD HARIS. NIM: 16521501**
- 4. KENNY BENAYA NATHAN. NIM 16521527**



**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2021**

DAFTAR ISI

DAFTAR ISI	1
BAB I	2
1.1 Dimensi, Besar File, Tipe, dan Sumber Dataset	2
1.2 Karakteristik Data	2
1.3 Data Cleansing	4
BAB II	6
2.1 Sampel Data	6
2.2 Statistik Atribut Data	6
2.4 Korelasi Data	14
BAB III	15
3.1 Diagram Review Summary	15
3.2. Grafik Current Players	16
3.3. Grafik Peak Players Today	16
3.4. Grafik Total Reviews	17
3.5. Diagram Scatter korelasi antara Current Players dan Peak Players Today	17
3.6. Diagram Scatter korelasi antara Current Players dan Total Review	18
3.7. Diagram Scatter korelasi antara Total Review dan Peak Players Today	19
3.8 Grafik Jumlah game Setiap Tahun	20
3.9 Grafik Genre per Tahun	20
BAB IV	29
4.1 Kesimpulan	29
4.2 Pembagian Tugas Kelompok	29

BAB I PENJELASAN DATASET

1.1 Dimensi, Besar File, Tipe, dan Sumber Dataset

1.1.1 Dimensi Dataset

Dataset yang digunakan memiliki ukuran 100 baris dan 8 kolom (100 x 8). Menggunakan pandas, hal ini didapatkan menggunakan perintah

```
df = pd.read_csv("steam_top_100.csv") #Define dataframe as df
len(df.columns) #hitung jumlah kolom
len(df) #hitung jumlah baris
```

1.1.2 Besar File Dataset

Besar file dataset ini adalah 27 KB

1.1.3 Tipe Dataset

Dataset ini adalah comma separated value (CSV) dengan pemisah berupa tanda koma (,)

1.1.4 Sumber Dataset

Dataset ini bersumber dari kaggle.com oleh username AngadChau dengan tautan <https://www.kaggle.com/angadchau/steam-top-100-gamesnov-2021>. Data diambil pada tanggal 15 November 2021.

1.2 Karakteristik Data

1.2.1 Penjelasan Kolom

Dataset ini terdiri dari 8 kolom. Kolom pertama adalah Steam ID. Fungsinya adalah sebagai pengenalan khusus dalam database yang digunakan oleh perusahaan Steam dalam mendata game yang ada di tokonya. Kolom kedua adalah Game yang berfungsi untuk menyimpan nama game yang didata di Steam. Kolom ketiga adalah Current Players yang menyimpan data berapa banyak orang yang sedang bermain game tersebut pada saat data diambil. Kolom keempat adalah Peak Players Today yang berfungsi untuk menjelaskan total pemain yang bermain pada hari itu. Kolom kelima adalah Release date adalah tanggal game tersebut dikeluarkan di Steam. Kolom keenam adalah review summary yang berisi penilaian game tersebut. Kolom ketujuh adalah kolom Total reviews yang berisi total penilaian pada satu game tersebut. Kolom kedelapan adalah kolom tags yang berisi tag untuk mencari game tersebut pada Steam.

1.2.2 Atribut Kolom

1.2.2.1 Kolom Current Players

Kolom Current Players adalah kolom dengan karakteristik data kuantitatif. Nilai maksimumnya adalah 736785 yaitu game Counter Strike: Global Offensive dan nilai minimumnya adalah 33310 yaitu game Battlefield V. Untuk mendapatkan data ini, diberikan perintah

```
#untuk data current players terbanyak
id_current_players_max = df["Current players"].idxmax() #Game dengan pemain
paling banyak
df[id_current_players_max:id_current_players_max+1]
```

Dengan hasil

Steam id	Game	Current players	Peak players today	Release date	Review summary	Total reviews	Tags
0	730 Counter-Strike: Global Offensive	736785	775266	21 Aug 2012	Very Positive	5993058.0	FPS:Shooter:Multiplayer:Competitive>Action:Tea...

Juga diberikan perintah

```
#untuk data current players paling sedikit
```

```
id_current_players_min = df["Current players"].idxmin() #Game dengan pemain
paling sedikit
df[id_current_players_min:id_current_players_min+1]
```

Dengan hasil

	Steam id	Game	Current players	Peak players today	Release date	Review summary	Total reviews	Tags
99	1238810	Battlefield™ V	6724	33310	22 Oct 2020	Mostly Positive	68109.0	FPS:World War II:Shooter:Multiplayer:War:Singl...

1.2.2.2 Kolom Peak Players Today

Kolom Peak players today adalah kolom dengan karakteristik data kuantitatif. Nilai maksimumnya adalah 775266 dengan game Counter Strike: Global Offensive. Nilai minimumnya adalah 8707 dengan game Assetto Corsa. Untuk mendapatkan hasil itu, dicari indexnya dengan perintah,

```
#data peak players today terbanyak
id_peak_players_max = df["Peak players today"].idxmax() #game dengan peak
players terbanyak
df[id_peak_players_max:id_peak_players_max+1]
```

Dengan hasil

	Steam id	Game	Current players	Peak players today	Release date	Review summary	Total reviews	Tags
0	730	Counter-Strike: Global Offensive	736875	775266	21 Aug 2012	Very Positive	5993058.0	FPS:Shooter:Multiplayer:Competitive>Action:Tea...

Juga diberikan perintah

```
#data peak players today paling sedikit
id_peak_players_min = df["Peak players today"].idxmin() #game dengan peak
players paling sedikit
df[id_peak_players_min:id_peak_players_min+1]
```

Dengan hasil

	Steam id	Game	Current players	Peak players today	Release date	Review summary	Total reviews	Tags
97	244210	Assetto Corsa	6965	8707	19 Dec 2014	Very Positive	42497.0	Racing:Automobile Sim:Simulation:Driving:Reali...

1.2.2.3 Kolom Total reviews

Kolom Total Reviews adalah kolom dengan karakteristik data kuantitatif. Nilai maksimumnya adalah 5993058 dengan game Counter Strike: Global Offensive dan minimumnya adalah 1433 dengan game Lost Ark Technical Beta. Untuk mendapatkan hasil tersebut, diberikan perintah,

```
#data total reviews terbanyak
id_reviews_max = df["Total reviews"].idxmax()
df[id_reviews_max:id_reviews_max+1]
```

Dengan hasil

	Steam id	Game	Current players	Peak players today	Release date	Review summary	Total reviews	Tags
0	730	Counter-Strike: Global Offensive	736875	775266	21 Aug 2012	Very Positive	5993058.0	FPS:Shooter:Multiplayer:Competitive>Action:Tea...

Juga diberikan perintah

```
#data total reviews paling sedikit
id_reviews_min = df["Total reviews"].idxmin()
df[id_reviews_min:id_reviews_min+1]
```

Juga diberikan perintah

	Steam id	Game	Current players	Peak players today	Release date	Review summary	Total reviews	Tags
13	1793660	Lost Ark Closed Technical Beta	46487	69408	29 Oct 2021	Overwhelmingly Positive	1433.0	NaN

1.2.2.4 Kolom Review Summary

Kolom review summary memiliki karakteristik kategorial data. Ada 4 kategori yaitu Very Positive, Overwhelmingly Positive, Mostly positive, dan mixed. Untuk mendapatkan data ini, diberikan perintah

```
df["Review summary"].value_counts() #Buat tabel frekuensi review summary
```

Hasilnya adalah

```
Very Positive      58
Overwhelmingly Positive  25
Mostly Positive    12
Mixed              4
Name: Review summary, dtype: int64
```

1.3 Data Cleansing

Dilakukan data cleansing untuk data 'Release Year'. Kelompok membuat kolom 'Release Year' dan mengambil 4 karakter terakhir dari kolom 'Release date' yang berupa tahun game dikeluarkan. Perintah yang digunakan adalah sebagai berikut,

```
#bikin kolom release year
df["Release Year"] = df["Release date"].str[-4:] #ambil 4 karakter terakhir
dari data release year
```

Dataframe yang dihasilkan adalah sebagai berikut,

```
Steam id  ... Release Year
0         730  ...      2012
1         570  ...      2013
2      1063730  ...      2021
3      1172470  ...      2020
4         440  ...      2007
..      ...  ...      ...
95     1286830  ...      2020
96     1644960  ...      2021
97      244210  ...      2014
98      275850  ...      2016
99     1238810  ...      2020

[100 rows x 9 columns]
```

Selain 'Release Year', dibuat juga kolom 'Genre' dengan cara membaca isi kolom tags dan mengisi kolom Genre 1 s.d. Genre 11 yang masing-masing kolom merepresentasikan genre top 100 games yang ada di Steam. Kolom Tags mencakup semua data karakteristik dari sebuah game, salah satunya genre game. Sebuah game dapat mencakup

lebih dari satu genre. Oleh karena itu, dibuat kolom Genre 1 hingga Genre 11 yang berisi karakteristik genre dari sebuah game. Cara mengambil data genre yang dimiliki oleh sebuah game adalah dengan mencari genre di suatu game di kolom Tags dengan algoritma yang ditunjukkan di bawah ini. Cara algoritma tersebut bekerja adalah dengan mencari dan menentukan terlebih dahulu apakah kolom Tags dari sebuah game mencakup list genre yang terdapat di situs resmi Steam Games. Kolom Genre 1 berisi apakah sebuah game termasuk ke dalam genre action. Kolom Genre 2 berisi apakah sebuah game termasuk ke dalam genre adventure. Kolom Genre 3 berisi apakah sebuah game termasuk ke dalam genre casual. Kolom Genre 4 berisi apakah sebuah game termasuk ke dalam genre experimental. Kolom Genre 5 berisi apakah sebuah game termasuk ke dalam genre puzzle. Kolom Genre 6 berisi apakah sebuah game termasuk ke dalam genre racing. Kolom Genre 7 berisi apakah sebuah game termasuk ke dalam genre RPG. Kolom Genre 8 berisi apakah sebuah game termasuk ke dalam genre simulation. Kolom Genre 9 berisi apakah sebuah game termasuk ke dalam genre sports. Kolom Genre 10 berisi apakah sebuah game termasuk ke dalam genre strategy. Kolom Genre 11 berisi apakah sebuah game termasuk ke dalam genre tabletop. Jika mencakup genre tersebut, baris di kolom genre yang bersangkutan akan diisi dengan nama genre tersebut. Di samping itu, jika sebuah game tidak mencakup genre tersebut, maka baris di kolom genre yang bersangkutan tersebut akan diisi “except (Genre yang sedang dicari keberadaannya)”, dengan except yang berarti “selain/di luar genre tersebut”. Perintah yang digunakan adalah sebagai berikut,

```
genre = ["Action", "Adventure", "Casual", "Experimental", "Puzzle", "Racing",
"RPG", "Simulation", "Sports", "Strategy", "Tabletop"] #list genre top-level
genres steam
df = df.astype({"Tags": str})
df["Genre 1"] = df["Tags"].map(lambda x: "Action" if "Action" in x else
"except Action")
df["Genre 2"] = df["Tags"].map(lambda x: "Adventure" if "Adventure" in x else
"except Adventure")
df["Genre 3"] = df["Tags"].map(lambda x: "Casual" if "Casual" in x else
"except Casual")
df["Genre 4"] = df["Tags"].map(lambda x: "Experimental" if "Experimental" in x
else "except Experimental")
df["Genre 5"] = df["Tags"].map(lambda x: "Puzzle" if "Puzzle" in x else
"except Puzzle")
df["Genre 6"] = df["Tags"].map(lambda x: "Racing" if "Racing" in x else
"except Racing")
df["Genre 7"] = df["Tags"].map(lambda x: "RPG" if "RPG" in x else "except
RPG")
df["Genre 8"] = df["Tags"].map(lambda x: "Simulation" if "Simulation" in x
else "except Simulation")
df["Genre 9"] = df["Tags"].map(lambda x: "Sports" if "Sports" in x else
"except Sports")
df["Genre 10"] = df["Tags"].map(lambda x: "Strategy" if "Strategy" in x else
"except Strategy")
df["Genre 11"] = df["Tags"].map(lambda x: "Tabletop" if "Tabletop" in x else
"except Tabletop")
```

Hasilnya adalah,

Genre 1	Genre 2	Genre 3	Genre 4	Genre 5	Genre 6	Genre 7	Genre 8	Genre 9	Genre 10
Action	except Adventure	except Casual	except Experimental	except Puzzle	except Racing	except RPG	except Simulation	Sports	Strategy
Action	except Adventure	except Casual	except Experimental	except Puzzle	except Racing	RPG	Simulation	Sports	Strategy
Action	Adventure	except Casual	except Experimental	except Puzzle	except Racing	RPG	except Simulation	except Sports	except Strategy
Action	except Adventure	except Casual	except Experimental	except Puzzle	except Racing	except RPG	except Simulation	except Sports	except Strategy
Action	except Adventure	except Casual	except Experimental	except Puzzle	except Racing	except RPG	except Simulation	except Sports	except Strategy

Data yang berisi except memiliki arti genre selain genre yang dimaksud.

BAB II STATISTIK DATA

2.1 Sampel Data

Sampel data yang diambil adalah beberapa game dengan jumlah pemain paling sedikit. Untuk itu diberikan perintah sebagai berikut

```
#mengambil 10 data game paling sedikit players
df_10_data = df.sort_values(["Current players"], ascending=[True])
df_10_data[0:10]
```

Dengan hasil

	Steam id	Game	Current players	Peak players today	Release date	Review summary	Total reviews	Tags
99	1238810	Battlefield™ V	6724	33310	22 Oct 2020	Mostly Positive	68109.0	FPS:World War II:Shooter:Multiplayer:War:Singl...
98	275850	No Man's Sky	6963	10122	12 Aug 2016	Mostly Positive	168792.0	Open World:Open World Survival Craft:Space:Exp...
97	244210	Assetto Corsa	6965	8707	19 Dec 2014	Very Positive	42497.0	Racing:Automobile Sim:Simulation:Driving:Real...
96	1644960	NBA 2K22	7020	13139	9 Sep 2021	Mixed	6266.0	Sports:Basketball:Simulation:eSports:Realistic...
95	1286830	STAR WARS™: The Old Republic™	7104	9345	21 Jul 2020	Very Positive	33098.0	Free to Play:MMORPG:Character Customization:Mu...
94	1184370	Pathfinder: Wrath of the Righteous	7182	10242	2 Sep 2021	Very Positive	9550.0	CRPG:Story Rich:Tactical RPG:RPG:Character Cus...
93	1293830	Forza Horizon 4	7606	9552	9 Mar 2021	Very Positive	69892.0	Racing:Open World:Driving:Multiplayer:Online C...
92	284160	BeamNG.drive	7874	10572	29 May 2015	Overwhelmingly Positive	85999.0	Simulation:Driving:Physics:Destruction:Automob...
91	242760	The Forest	7918	11706	30 Apr 2018	Very Positive	264327.0	Open World Survival Craft:Survival:Open World:...
90	632360	Risk of Rain 2	8220	11724	11 Aug 2020	Overwhelmingly Positive	112385.0	Third-Person Shooter>Action Roguelike:Multipla...

2.2 Statistik Atribut Data

2.2.1 Current players

2.2.1.1 Rata - rata dan Standar Deviasi

Dari kolom Current players yang bertipe data kuantitatif, maka rata - rata dapat dicari dengan memasukkan perintah

```
df["Current players"].mean() #Mencari nilai rata-rata
```

Dengan hasil

```
39954.37
```

Sedangkan untuk standar deviasi dapat dicari dengan menggunakan perintah

```
df["Current players"].std() #Mencari standar deviasi
```

Hasilnya

```
94808.09286149277
```

2.2.1.1 Percentile

2.2.1.1.1 Percentile 10%

Pada kolom Current Players, percentile 10% dapat dicari dengan menggunakan perintah

```
df["Current players"].quantile(0.1) #Persentil 10%
```


Dengan hasil sebagai berikut

```
8239.8
```

2.2.1.1.2 Percentile 25%

Pada kolom Current players, percentile 25% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Current players"].quantile(0.25) #Persentil 25%
```

Dengan hasil

```
10018.5
```

2.2.1.1.3 Percentile 50%

Pada kolom Current players, percentile 50% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Current players"].quantile(0.5) #Persentil 50%
```

Dengan hasil

```
18012.0
```

2.2.1.1.4 Percentile 75%

Pada kolom Current players, percentile 75% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Current players"].quantile(0.75) #Persentil 75%
```

Dengan hasil

```
32333.5
```

2.2.1.1.5 Percentile 90%

Pada kolom Current players, percentile 90% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Current players"].quantile(0.9) #Persentil 90%
```

Dengan hasil

```
55296.9000000000016
```

2.2.1.3 Ekstremum

2.2.1.3.1 Nilai Maksimum

Dari kolom Current players yang bertipe kuantitatif, maka nilai maksimum dari Current players dapat didapatkan dengan menggunakan perintah

```
df["Current players"].max() #Mencari nilai maksimum
```

Dengan hasil

```
736875
```

Jika menggunakan perintah,

```
imax=df["Current players"].idxmax() # Mencari indeks dengan nilai maksimum  
pada atribut Current players  
df[imax:imax+1]
```

Didapatkan bahwa game dengan current players terbanyak adalah game Counter Strike: Global Offensive

2.2.1.3.2 Nilai Minimum

Nilai minimum dari kolom Current players dapat dicari dengan perintah

```
df["Current players"].min() #Mencari nilai minimum
```

Hasilnya

```
6724
```

Jika menggunakan perintah,

```
imin=df["Current players"].idxmin() # Mencari indeks dengan nilai minimum pada  
atribut Current players  
df[imin:imin+1]
```

Akan didapatkan bahwa game dengan current players paling sedikit adalah game Battlefield V.

2.2.2 Peak Players Today

2.2.2.1 Rata - rata dan Standar Deviasi

Dari kolom Peak players today yang bertipe data kuantitatif, maka rata - rata dapat dicari dengan memasukkan perintah

```
df["Peak players today"].mean() #Mencari nilai rata-rata
```

Dengan hasil

```
52364.19
```

Sedangkan untuk standar deviasi dapat dicari dengan menggunakan perintah

```
df["Peak players today"].std() #Mencari Standar Deviasi
```

Hasilnya

```
106888.74481853891
```

2.2.2.2 Percentile

2.2.2.2.1 Percentile 10%

Pada kolom Peak players today, percentile 10% dapat dicari dengan menggunakan perintah

```
df["Peak players today"].quantile(0.1) #Persentil 10%
```

Dengan hasil sebagai berikut

```
10678.2
```

2.2.2.2.2 Percentile 25%

Pada kolom Peak players today, percentile 25% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Peak players today"].quantile(0.25) #Persentil 25%
```

Dengan hasil

```
15032.75
```

2.2.2.2.3 Percentile 50%

Pada kolom Peak players today, percentile 50% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Peak players today"].quantile(0.5) #Persentil 50%
```

Dengan hasil

```
24493.0
```

2.2.2.2.4 Percentile 75%

Pada kolom Peak players today, percentile 75% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Peak players today"].quantile(0.75) #Persentil 75%
```

Dengan hasil

```
43076.0
```

2.2.2.2.5 Percentile 90%

Pada kolom Peak players today, percentile 90% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Peak players today"].quantile(0.9) #Persentil 90%
```

Dengan hasil

```
71377.600000000011
```

2.2.2.3 Ekstremum

2.2.2.3.1 Nilai Maksimum

Dari kolom Current players yang bertipe kuantitatif, maka nilai maksimum dari Current players dapat didapatkan dengan menggunakan perintah

```
df["Peak players today"].max() #Mencari nilai maksimum
```

Dengan hasil

```
775266
```

Menggunakan perintah,

```
imax=df["Peak players today"].idxmax() # Mencari indeks dengan nilai maksimum  
pada atribut Peak Players Today  
df[imax:imax+1]
```

Akan didapatkan bahwa Peak Players Today terbanyak adalah game Counter Strike: Global Offensive.

2.2.2.3.2 Nilai Minimum

Nilai minimum dari kolom Current players dapat dicari dengan perintah

```
df["Peak players today"].min() #Mencari nilai minimum
```

Hasilnya

```
8707
```

Menggunakan perintah,

```
imin=df["Peak players today"].idxmin() # Mencari indeks dengan nilai minimum  
pada atribut Peak players today  
df[imin:imin+1]
```

Akan didapatkan bahwa game dengan Peak players paling sedikit adalah game Assetto Corsa.

2.2.3 Total Reviews

2.2.3.1 Rata - rata dan Standar Deviasi

Dari kolom Total Reviews yang bertipe data kuantitatif, maka rata - rata dapat dicari dengan memasukkan perintah

```
df["Total reviews"].mean() #Mencari nilai rata-rata
```

Dengan hasil

```
269622.0404040404
```

Sedangkan untuk standar deviasi dapat dicari dengan menggunakan perintah

```
df["Total reviews"].std() #Mencari standar deviasi
```

Hasilnya

```
644753.9116125565
```

2.2.3.2 Percentile

2.2.3.2.1 Percentile 10%

Pada kolom Total reviews, percentile 10% dapat dicari dengan menggunakan perintah

```
df["Total reviews"].quantile(0.1) #Persentil 10%
```

Dengan hasil sebagai berikut

```
15107.000000000002
```

2.2.3.2.2 Percentile 25%

Pada kolom Total reviews, percentile 25% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Total reviews"].quantile(0.25) #Persentil 25%
```

Dengan hasil

```
48593.5
```

2.2.3.2.3 Percentile 50%

Pada kolom Total reviews, percentile 50% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Total reviews"].quantile(0.5) #Persentil 50%
```

Dengan hasil

```
111163.0
```

2.2.3.2.4 Percentile 75%

Pada kolom Total reviews, percentile 75% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Total reviews"].quantile(0.75) #Persentil 75%
```

Dengan hasil

```
274705.0
```

2.2.3.2.5 Percentile 90%

Pada kolom Total reviews, percentile 90% dapat dicari dengan menggunakan perintah sebagai berikut

```
df["Total reviews"].quantile(0.9) #Persentil 90%
```

Dengan hasil

```
466846.40000000002
```

2.2.3.3 Ekstremum

2.2.3.3.1 Nilai Maksimum

Dari kolom Total reviews yang bertipe kuantitatif, maka nilai maksimum dari Current players dapat didapatkan dengan menggunakan perintah

```
df["Total reviews"].max() #Mencari nilai maksimum
```

Dengan hasil

```
5993058.0
```

Menggunakan perintah,

```
imax=df["Total reviews"].idxmax() # Mencari indeks dengan nilai maksimum pada  
atribut Total Reviews  
df[imax:imax+1]
```

Didapatkan bahwa penilaian terbanyak ada pada game Counter Strike: Global Offensive

2.2.3.3.2 Nilai Minimum

Nilai minimum dari kolom Total reviews dapat dicari dengan perintah

```
df["Total reviews"].min() #Mencari nilai minimum
```

Hasilnya

```
1433.0
```

Menggunakan perintah,

```
imin=df["Total reviews"].idxmin() # Mencari indeks dengan nilai maksimum pada  
atribut Total Reviews  
df[imin:imin+1]
```

Akan didapatkan bahwa game dengan penilaian paling sedikit adalah game Lost Ark Closed Technical Beta

2.3 Tabel Frekuensi

Setelah data cleansing, dataset memiliki 9 kolom dan yang dapat dikonversi menjadi tabel frekuensi adalah kolom 'Review summary' dan 'Release Year'.

2.3.1 Review Summary

Kolom ini memuat kategori beberapa review dari pengguna yang dibedakan menjadi 4 jenis yaitu Very Positive, Overwhelmingly Positive, Mostly Positive, dan Mixed. Dari beberapa kategori tersebut akan dibuat tabel frekuensi yang memuat jumlah tiap kategori reviewnya yang dapat dicari dengan menggunakan perintah

```
df["Review summary"].value_counts() #Buat tabel frekuensi review summary
```

Hasilnya adalah

```
Very Positive          58
Overwhelmingly Positive 25
Mostly Positive        12
Mixed                  4
Name: Review summary, dtype: int64
```

Dari data ini, disimpulkan bahwa untuk game yang masuk top 100 Steam sebagian besar memiliki penilaian 'Very Positive'.

2.3.2 Release Year

Kolom ini memuat tahun game dikeluarkan dengan game terbaru dirilis tahun 2021 dan paling lama dirilis tahun 2001. Tabel frekuensi didapatkan dengan cara,

```
df["Release Year"].value_counts() #Buat tabel frekuensi release year
```

Hasilnya adalah

```
2020    18
2021    16
2018     11
2017     11
2016     10
2015      8
2013      8
2019      5
2014      4
2012      2
2006      1
2009      1
2011      1
2010      1
2007      1
2000      1
Name: Release Year, dtype: int64
```

Dari data ini, disimpulkan bahwa game yang masuk pada top 100 Steam sebagian besar dikeluarkan pada tahun 2020 dan paling tua dikeluarkan pada tahun 2000.

2.4 Korelasi Data

Korelasi data didapatkan menggunakan fungsi `.corr()`. Jika hasil dari perintah `.corr()` nilainya mendekati 1, data tersebut artinya saling berkorelasi. Jika hasilnya negatif, data tersebut tidak saling berkorelasi.

2.4.1 Korelasi data Current players dengan Peak Players Today

```
df["Current players"].corr(df["Peak players today"]) #cari korelasi antara  
current players dengan peak players today
```

Hasilnya

```
0.9775687581213371
```

Karena hasilnya mendekati angka 1, artinya Current players berkorelasi dengan Peak players today.

2.4.2 Korelasi data Current players dengan Total Reviews

```
df["Current players"].corr(df["Total reviews"]) #cari korelasi antara current  
players dengan peak players today
```

Hasilnya

```
0.8478876941579045
```

Karena hasilnya mendekati angka 1, artinya Current players berkorelasi dengan Total Reviews.

2.4.3 Korelasi data Peak Players Today dengan Total Reviews

```
df["Current players"].corr(df["Total reviews"]) #cari korelasi antara current  
players dengan peak players today
```

Hasilnya

```
0.8478876941579045
```

Karena hasilnya mendekati angka 1, artinya Current players berkorelasi dengan Total Reviews.

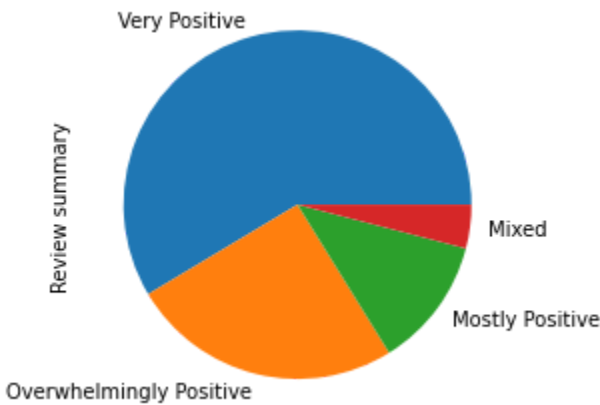
BAB III VISUALISASI DATA

3.1 Diagram Review Summary

Menggunakan perintah,

```
pie=df["Review summary"].value_counts()  
pie.plot(kind="pie")  
plt.show()
```

Didapatkan *pie chart*,



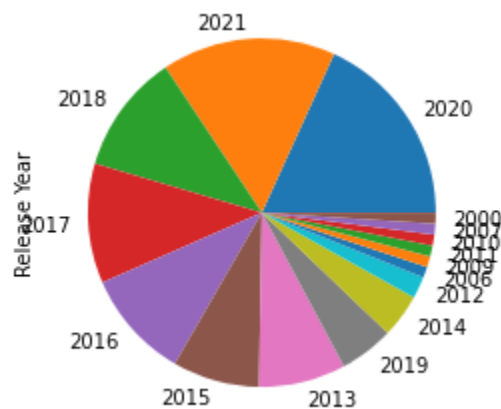
Dari *pie chart* tersebut, diketahui bahwa game yang ada di list 100 game Steam sebagian besar memiliki review “Very Positive”. Artinya, sebagian besar 100 game tersebut adalah game yang baik

3.2 Diagram Release Year

Menggunakan perintah,

```
pie = df["Release Year"].value_counts()  
pie.plot(kind="pie")  
plt.show()
```

Didapatkan *pie chart*,



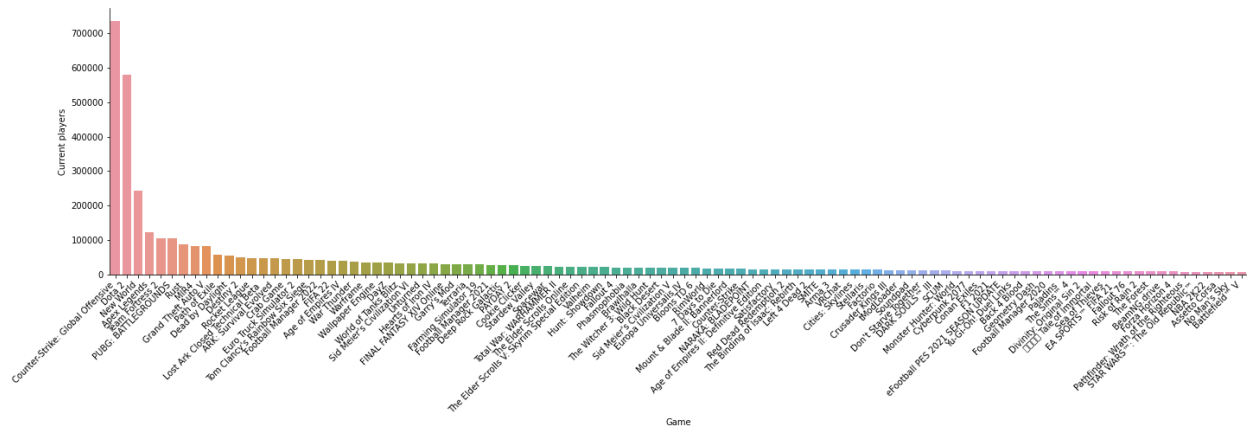
Dari *pie chart* tersebut, dapat dilihat bahwa sebagian besar game dirilis pada dekade ini (2010-2021). Tahun 2020 adalah tahun dengan game terbanyak yang dirilis, diikuti dengan tahun 2021.

3.2. Grafik Current Players

Dikarenakan data yang cukup banyak, kelompok menggunakan seaborn untuk membuat grafik (untuk selanjutnya, seaborn diimport menggunakan perintah `import seaborn as sns`). Menggunakan perintah,

```
sns.catplot(x = 'Game', y = 'Current players', data = df, kind = 'bar',
linewidth = 0.5, aspect=4)
plt.xticks(rotation = 45, ha = 'right')
```

Didapatkan grafik bar,



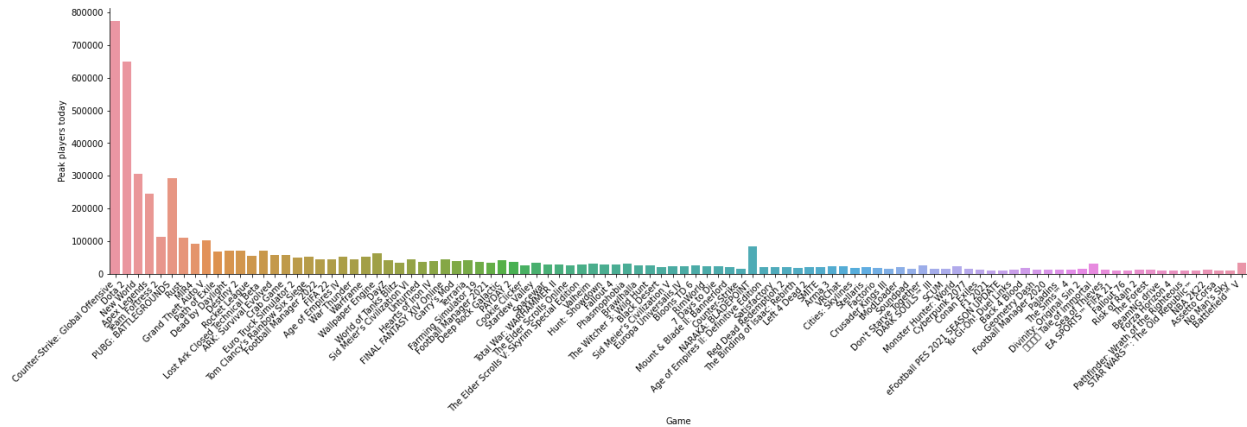
Dari grafik bar tersebut, diketahui bahwa game Counter Strike: Global Offensive memiliki jumlah player terbanyak dan game Battlefield V memiliki jumlah current player paling sedikit.

3.3. Grafik Peak Players Today

Dikarenakan data yang cukup banyak, kelompok menggunakan seaborn untuk membuat grafik. Menggunakan perintah,

```
sns.catplot(x = 'Game', y = 'Peak players today', data = df, kind = 'bar',
linewidth = 0.5, aspect=4)
plt.xticks(rotation = 45, ha = 'right')
```

Didapatkan grafik bar,



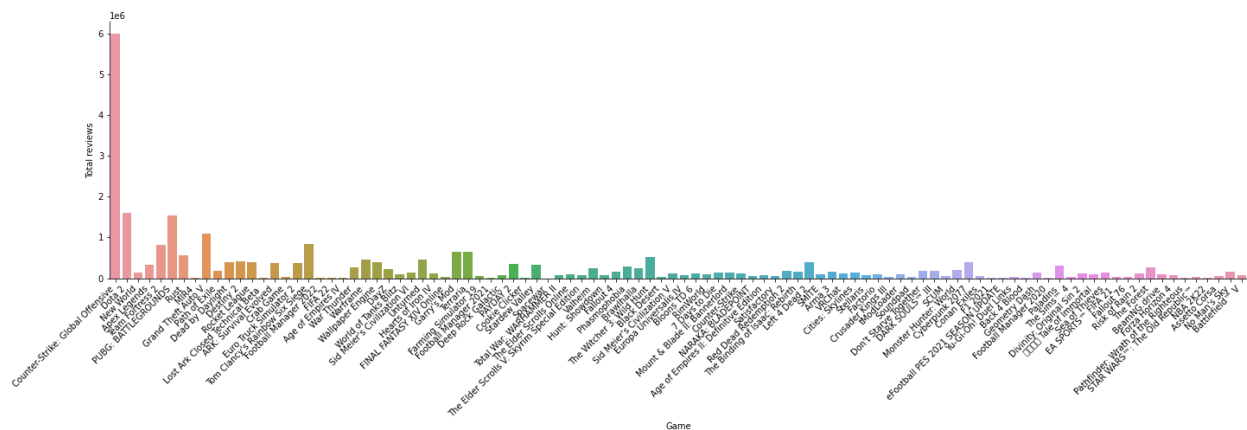
Dari grafik bar tersebut, diketahui bahwa game Counter Strike: Global Offensive memiliki jumlah Peak Players Today terbanyak dan game Assetto Corsa memiliki jumlah Peak Players Today paling sedikit.

3.4. Grafik Total Reviews

Dikarenakan data yang cukup banyak, kelompok menggunakan seaborn untuk membuat grafik. Menggunakan perintah,

```
sns.catplot(x = 'Game', y = 'Total reviews',data = df, kind = 'bar', linewidth = 0.5, aspect=4)
plt.xticks(rotation = 45, ha = 'right')
```

Didapatkan grafik bar,



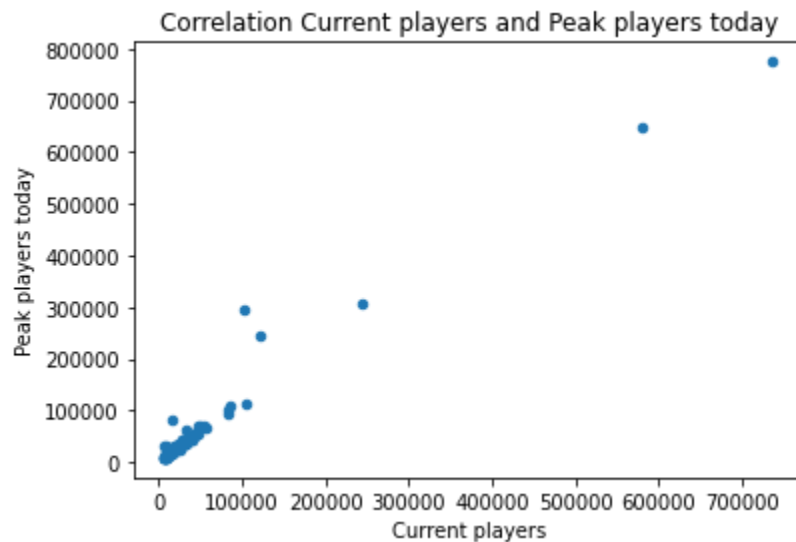
Dari grafik bar tersebut, diketahui bahwa game Counter Strike: Global Offensive memiliki total review yang terbanyak dan game Lost Ark Technical Beta memiliki total review paling sedikit.

3.5. Diagram Scatter korelasi antara Current Players dan Peak Players Today

Menggunakan perintah,

```
df.plot(kind="scatter", x="Current players", y="Peak players today",  
title="Correlation Current players and Peak players today")  
plt.show()
```

Didapatkan *Scatter Plot*,



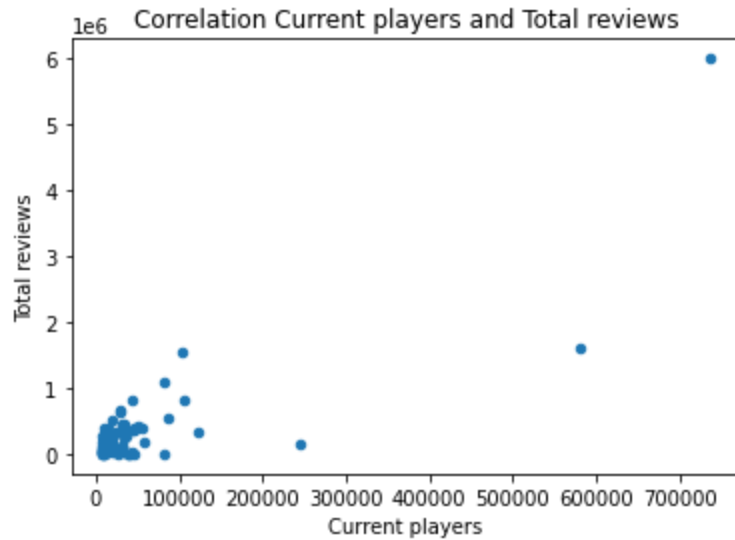
Dikarenakan ada bentuk garis pada data awal, maka *scatter plot* ini memiliki hubungan yang positif dengan kata lain, jika current players bertambah maka peak players today juga bertambah.

3.6. Diagram Scatter korelasi antara Current Players dan Total Review

Menggunakan perintah,

```
df.plot(kind="scatter", x="Current players", y="Total reviews",  
title="Correlation Current players and Total reviews")  
plt.show()
```

Didapatkan *Scatter Plot*,



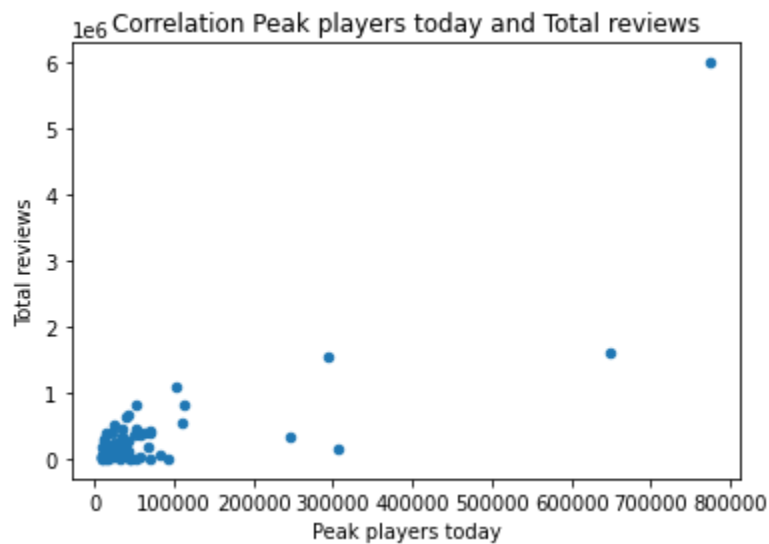
Pada *scatter plot* antara Current Players dan Total Review, adanya pengumpulan data di area yang jumlah Total Review serta Current Players cukup sedikit. Setelah itu, di angka sekitar 100.000 Current Players, ada sedikit penyebaran data. Ada pula sebuah data Outlier pada angka Total Review dan Current Players yang besar.

3.7. Diagram Scatter korelasi antara Total Review dan Peak Players Today

Menggunakan perintah,

```
df.plot(kind="scatter", x="Peak players today", y="Total reviews",
title="Correlation Peak players today and Total reviews")
plt.show()
```

Didapatkan *Scatter Plot*,



Scatter plot antara Total Review dan Peak Players Today mirip dengan *scatter plot* antara Current Players dan Total Reviews, di mana ada pengumpulan data pada area yang jumlah antara Total Review ataupun Peak Players Today

sedikit. Setelah itu, semakin besar Peak Players today, datanya juga menunjukkan adanya penyebaran. *Scatter plot* ini juga menunjukkan adanya sebuah outlier atau data yang perbedaannya cukup ekstrim dari pola data-data lainnya.

3.8 Grafik Jumlah Game Setiap Tahun

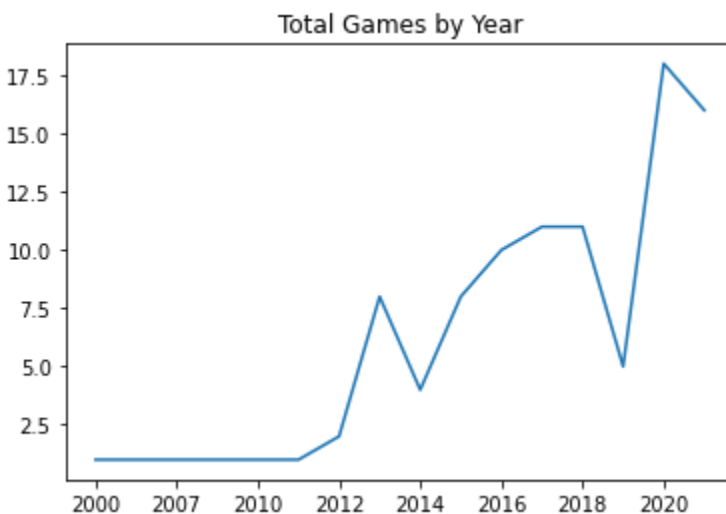
Untuk mendapatkan jumlah game setiap tahunnya, dibuat dataframe 2 dengan perintah,

```
df2 = df["Release Year"].value_counts().sort_index() #Line Chart Total Games by Year
```

Dari df2, dibuat grafiknya dengan perintah,

```
df2.plot(kind="line", x="Release Year", y="counts", title="Total Games by Year")
```

Hasilnya adalah,



Dari grafik ini, ditunjukkan bahwa game yang masuk urutan *top 100* Steam sebagian besar dikeluarkan pada tahun 2020 dan game paling tua yang masuk urutan *top 100* dikeluarkan pada tahun 2000

3.9 Grafik Genre per Tahun

Akan dibuat grafik untuk setiap genre setiap tahunnya. Diambil 11 genre dari kolom tags menggunakan perintah,

```
genre = ["Action", "Adventure", "Casual", "Experimental", "Puzzle", "Racing", "RPG", "Simulation", "Sports", "Strategy", "Tabletop"] #list genre top-level genres steam
df["Release Year"] = df["Release date"].str[-4:] #ambil 4 karakter terakhir dari data release year
df = df.astype({"Tags": str})
df["Genre 1"] = df["Tags"].map(lambda x: "Action" if "Action" in x else "except Action")
df["Genre 2"] = df["Tags"].map(lambda x: "Adventure" if "Adventure" in x else "except Adventure")
df["Genre 3"] = df["Tags"].map(lambda x: "Casual" if "Casual" in x else
```

```

"except Casual")
df["Genre 4"] = df["Tags"].map(lambda x: "Experimental" if "Experimental" in x
else "except Experimental")
df["Genre 5"] = df["Tags"].map(lambda x: "Puzzle" if "Puzzle" in x else
"except Puzzle")
df["Genre 6"] = df["Tags"].map(lambda x: "Racing" if "Racing" in x else
"except Racing")
df["Genre 7"] = df["Tags"].map(lambda x: "RPG" if "RPG" in x else "except
RPG")
df["Genre 8"] = df["Tags"].map(lambda x: "Simulation" if "Simulation" in x
else "except Simulation")
df["Genre 9"] = df["Tags"].map(lambda x: "Sports" if "Sports" in x else
"except Sports")
df["Genre 10"] = df["Tags"].map(lambda x: "Strategy" if "Strategy" in x else
"except Strategy")
df["Genre 11"] = df["Tags"].map(lambda x: "Tabletop" if "Tabletop" in x else
"except Tabletop")
print(df)

```

Dari perintah tersebut, dibuat grafik untuk sebelas genre tersebut.

3.9.1 Grafik Action Games setiap Tahunnya

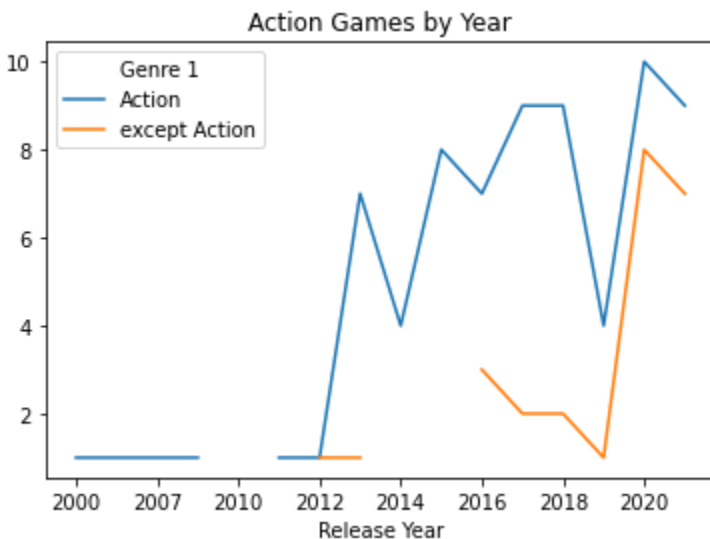
Grafik dibuat dengan perintah,

```

df.groupby(["Release Year", "Genre 1"]).size().unstack().plot(title="Action
Games by Year") #x="Release Year", y="Genre 1"

```

Menghasilkan grafik,



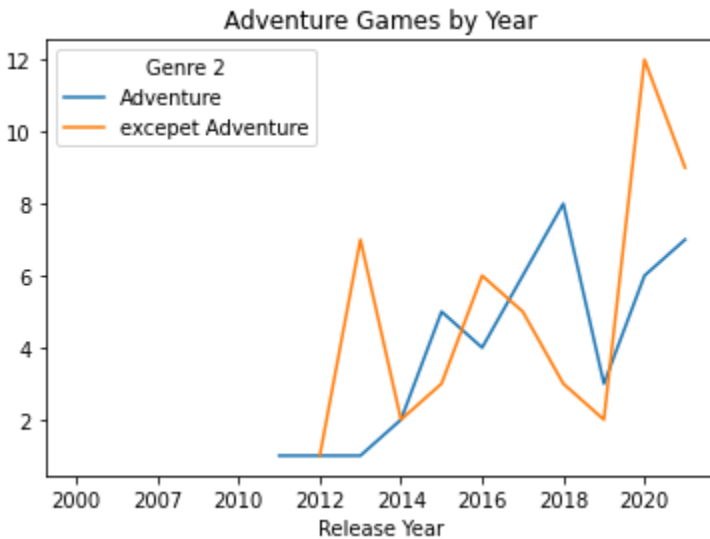
Dari grafik tersebut, dapat terlihat jumlah action games yang dirilis serta masuk top 100 games di Steam, cukup banyak dibandingkan dengan game dengan genre lain pada top 100 Games di Steam. Selain itu, dalam jangka waktu yang relatif panjang, grafik game dengan genre action yang dirilis menunjukkan peningkatan yang cukup besar. Hal tersebut menunjukkan bahwa game dengan genre action sangat diminati oleh gamers. Tahun 2020 merupakan tahun dengan jumlah Action Games terbanyak yang dirilis dan ada pada top 100 games di Steam.

3.9.2 Grafik Adventure Games setiap Tahunnya

Grafik dibuat dengan perintah

```
df.groupby(["Release Year", "Genre 2"])["Game"].size().unstack().plot(title="Adventure Games by Year")#x=Release Year,y="Genre 2"
```

Menghasilkan grafik,



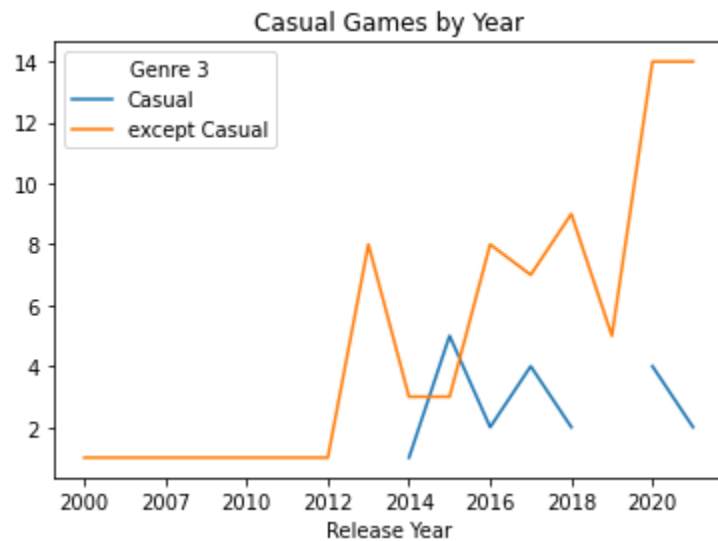
Dari grafik ini, dapat terlihat bahwa dalam jangka waktu yang relatif panjang, grafik game dengan genre adventure yang dirilis menunjukkan peningkatan yang baik. Hal tersebut menunjukkan bahwa game dengan genre adventure sangat diminati oleh gamers. Selain itu, dapat diketahui bahwa tahun 2018 merupakan tahun dengan jumlah Adventure Games terbanyak yang dirilis dan masuk ke top 100 Games di Steam

3.9.3 Grafik Casual Games setiap tahunnya

Grafik dihasilkan dengan perintah,

```
df.groupby(["Release Year", "Genre 3"])["Game"].size().unstack().plot(title="Casual Games by Year")#x="Release Year",y="Genre 3"
```


Menghasilkan grafik,



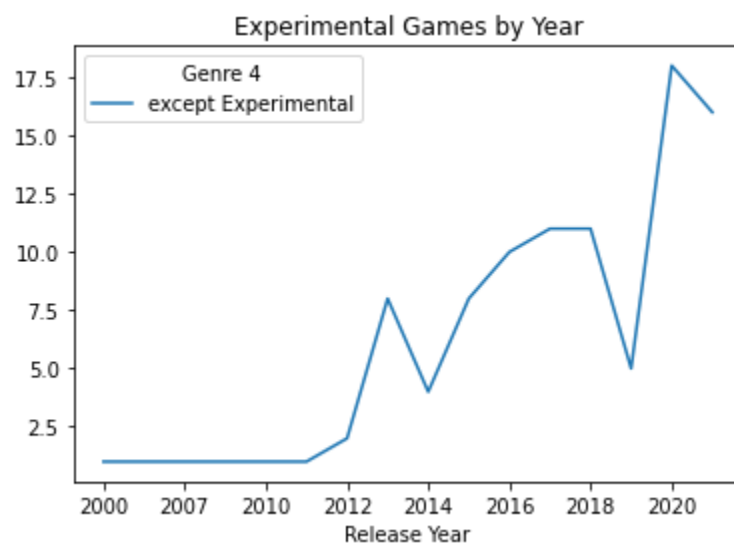
Dari grafik ini, dapat dilihat jumlah Casual games terbanyak yang dirilis dan masuk top 100 Games di Steam adalah pada tahun 2015. Selain itu, dalam jangka waktu yang relatif panjang, dapat dilihat bahwa grafik casual games yang dirilis cenderung stagnan atau tidak menunjukkan peningkatan maupun penurunan yang signifikan. Hal tersebut menunjukkan bahwa casual games cukup diminati oleh masyarakat, karena genre game tersebut masih termasuk ke dalam top 100 steam games.

3.9.4 Grafik Experimental Games setiap tahunnya

Grafik dihasilkan dengan perintah,

```
df.groupby(["Release Year", "Genre 4"])["Game"].size().unstack().plot(title="Experimental Games by Year") #x="Release Year", y="Genre 4"
```

Menghasilkan grafik,



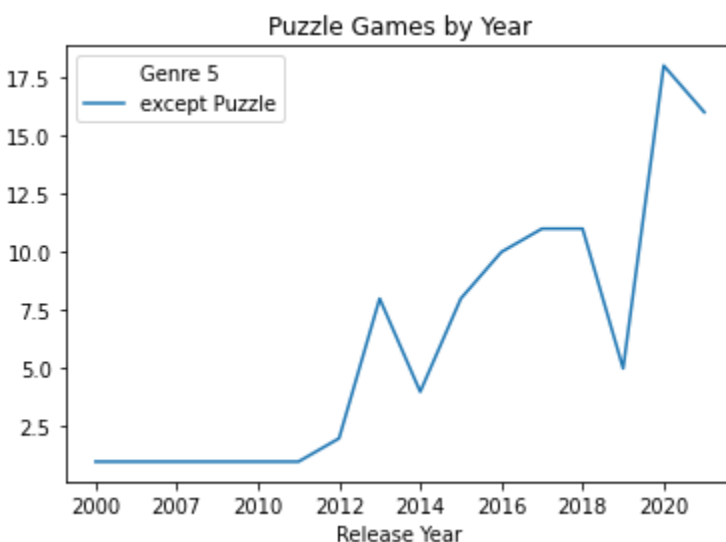
Dari grafik tersebut, tidak terlihat data dari game dengan genre experimental. Artinya, dapat disimpulkan bahwa dalam top 100 steam games yang dirilis pada tahun 2000 hingga 2021, tidak ada game dengan genre experimental. Hal tersebut dapat memberi indikasi bahwa game dengan genre experimental tidak terlalu diminati oleh gamers karena game dengan genre experimental yang dirilis tidak termasuk ke dalam top 100 steam games.

3.9.5 Grafik Puzzle Games setiap tahunnya

Grafik dihasilkan dengan perintah,

```
df.groupby(["Release Year", "Genre
5"]) ["Game"].size().unstack().plot(title="Puzzle Games by Year") #x="Release
Year",y="Genre 5"
```

Menghasilkan grafik,



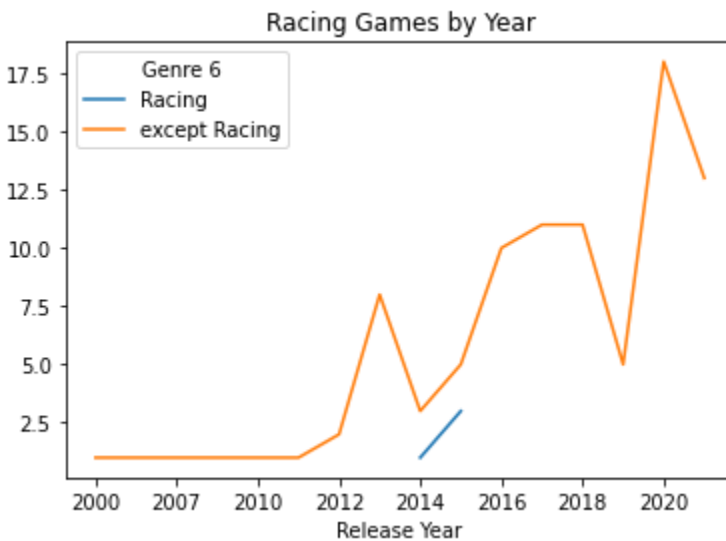
Dari grafik ini, disimpulkan bahwa tidak ada puzzle games yang masuk pada top 100 game steam pada tahun 2000 hingga 2021. Hal tersebut menunjukkan bahwa game dengan genre puzzle tidak terlalu diminati oleh gamers karena puzzle games yang dirilis tidak termasuk ke dalam top 100 steam games.

3.9.6 Grafik Racing Games setiap tahunnya

Grafik dihasilkan dengan perintah,

```
df.groupby(["Release Year", "Genre
6"]) ["Game"].size().unstack().plot(title="Racing Games by Year") #x="Release
Year",y="Genre 6"
```

Menghasilkan grafik,



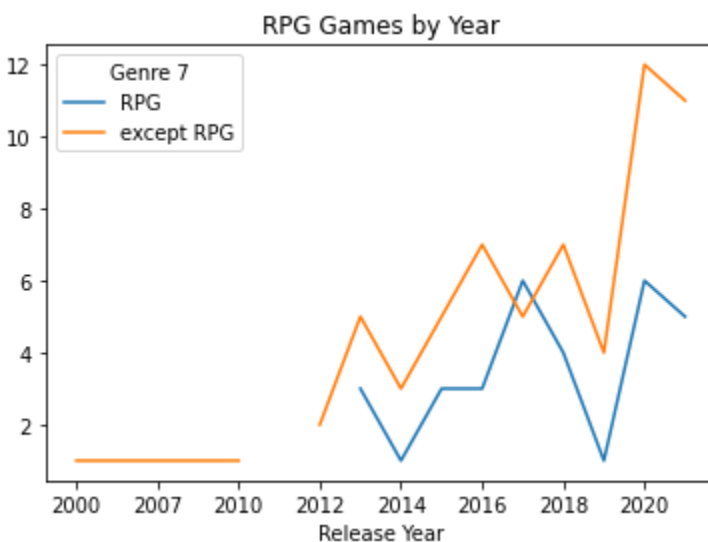
Dari grafik ini, disimpulkan bahwa game genre racing hanya dirilis pada tahun 2014-2015. Game dengan genre racing juga hanya mencakup sebagian kecil dari keseluruhan genre games yang dirilis. Oleh karena itu, game dengan genre racing tidak terlalu diminati gamers.

3.9.7 Grafik RPG Games setiap tahunnya

Grafik dihasilkan dengan perintah,

```
df.groupby(["Release Year", "Genre 7"])["Game"].size().unstack().plot(title="RPG Games by Year") #x="Release Year", y="Genre 7"
```

Menghasilkan grafik,



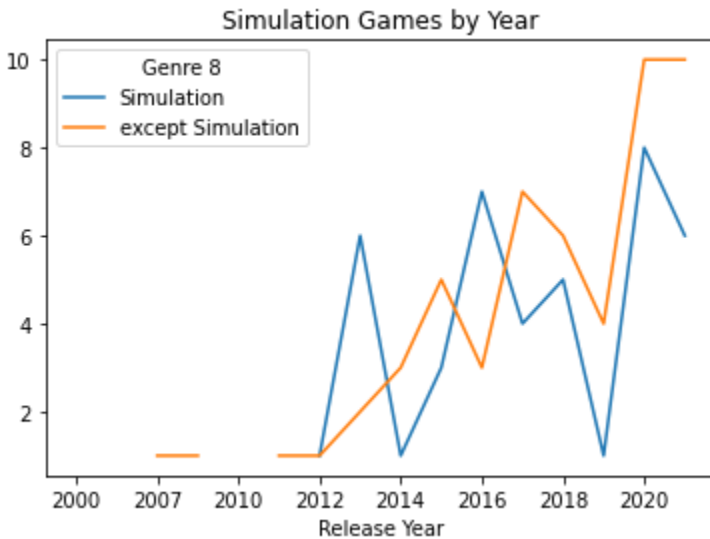
Dari grafik diatas, dapat terlihat bahwa pada beberapa tahun terakhir ini grafik game dengan genre RPG yang dirilis cenderung meningkat dengan cukup baik dalam waktu yang relatif panjang. Hal tersebut mengindikasikan bahwa game dengan genre RPG cukup diminati oleh gamer.

3.9.8 Grafik Simulation Games setiap tahunnya

Grafik dihasilkan dengan perintah,

```
df.groupby(["Release Year", "Genre 8"])["Game"].size().unstack().plot(title="Simulation Games by Year") #x="Release Year", y="Genre 8"
```

Menghasilkan grafik,



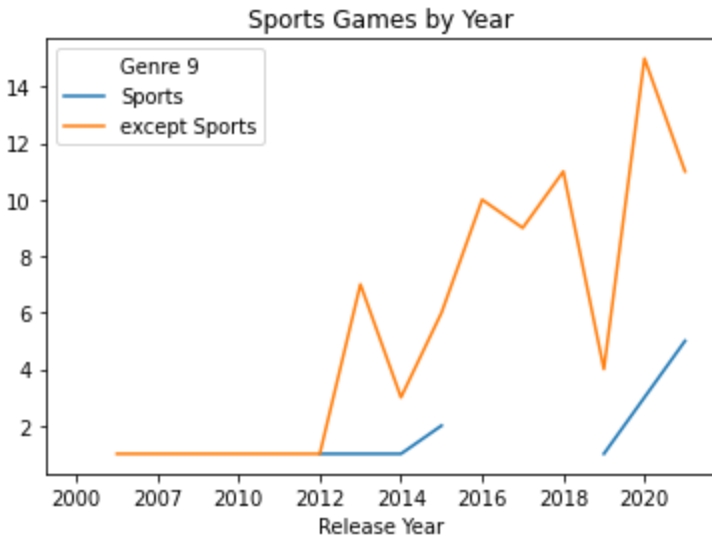
Pada grafik diatas, dapat dilihat bahwa pada beberapa tahun belakang game dengan genre simulation selalu dirilis di steam di tiap tahunnya dan bahkan cenderung meningkat dari tahun ke tahun. Oleh karena itu, games dengan genre simulation diminati oleh gamers.

3.9.9 Grafik Sports Games setiap tahunnya

Grafik dihasilkan dengan perintah,

```
df.groupby(["Release Year", "Genre 9"])["Game"].size().unstack().plot(title="Sports Games by Year") #x="Release Year", y="Genre 9"
```

Menghasilkan grafik,



Pada grafik diatas, dapat dilihat bahwa terdapat game dengan genre sports namun dengan porsi yang cukup kecil pada top 100 steam games. Hal ini menandakan game dengan genre sport kurang memiliki daya tarik pada pemain atau sports games tidak terlalu diminati gamers.

3.9.10 Grafik Strategy Games setiap tahunnya

Grafik dihasilkan dengan perintah,

```
df.groupby(["Release Year", "Genre 10"])["Game"].size().unstack().plot(title="Strategy Games by Year") #x="Release Year", y="Genre 10"
```

Menghasilkan grafik,



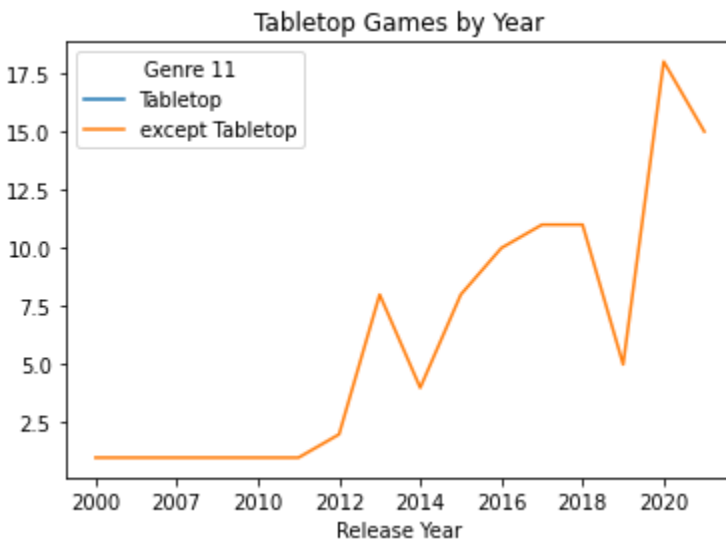
Dari grafik diatas, dapat dilihat bahwa pada beberapa tahun terakhir, perilsan game dengan genre strategy terus ada dan cenderung stabil. Hal ini menunjukkan bahwa game dengan genre strategy mendapat respon yang cukup positif dari para pemain.

3.9.11 Grafik Tabletop Games setiap tahunnya

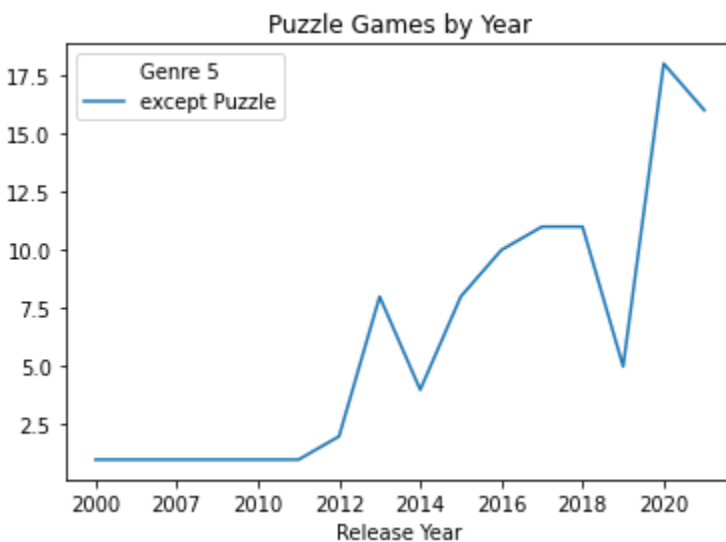
Grafik dihasilkan dengan perintah,

```
df.groupby(["Release Year", "Genre  
11"]) ["Game"].size().unstack().plot(title="Tabletop Games by Year")  
#x="Release Year", y="Genre"
```

Menghasilkan grafik,



Dari grafik ini, disimpulkan bahwa game dengan genre table top hanya pernah dirilis dalam jumlah yang sangat kecil. Hal tersebut karena sistem algoritma pencari genre masih mendeteksi keberadaan game dengan genre tabletop seperti yang ditunjukkan di garis biru di atas. Jika sebuah genre tidak terdapat dalam top 100 steam games, maka gambar grafik yang ditampilkan hanya akan memberi informasi mengenai garis “except (Genre yang dicari)” seperti pada gambar grafik di bawah ini.



Game dengan genre tabletop terdapat pada index 94 sebagaimana ditampilkan di gambar di bawah ini.

```

  Steam id      Game ...      Genre 10 Genre 11
94  1184370  Pathfinder: Wrath of the Righteous ... except Strategy Tabletop

[1 rows x 20 columns]

```

Dengan jumlah game dengan genre tabletop pada top 100 steam games yang sangat sedikit, maka dapat disimpulkan bahwa game dengan genre tabletop tidak terlalu diminati gamers.

BAB IV KESIMPULAN

4.1 Kesimpulan

Kesimpulan dari pengerjaan tugas besar ini adalah,

1. Pandas dapat digunakan untuk menganalisis data dan mendeskripsikannya
2. Matplot dapat melakukan visualisasi data dari data yang diberikan
3. Penyimpanan data besar dapat menggunakan file csv dengan pemisah yang berbeda-beda (, atau ;)
4. Pemilihan visualisasi data sangat berpengaruh terhadap pemahaman pembaca

4.2 Pembagian Tugas Kelompok

Pembagian tugas besar ini adalah sebagai berikut,

1. Matthew Mahendra (16521497): Dimensi, Besar, dan Ukuran Dataset, Karakteristik Data, Data Cleansing
2. Henry Anand Septian Radityo (16521507): Visualisasi Data, Mencari Dataset, Merekam
3. Richard Haris (16521501): Visualisasi Data, Statistik Data, Dimensi, Besar dan Ukuran Dataset, Data Cleansing
4. Kenny Benaya Nathan (16521527): Statistik Data, Membuat Presentasi, Visualisasi Data