A/B TESTING PROJECT

MARCH 2

Ferdyansyah Permana Putra

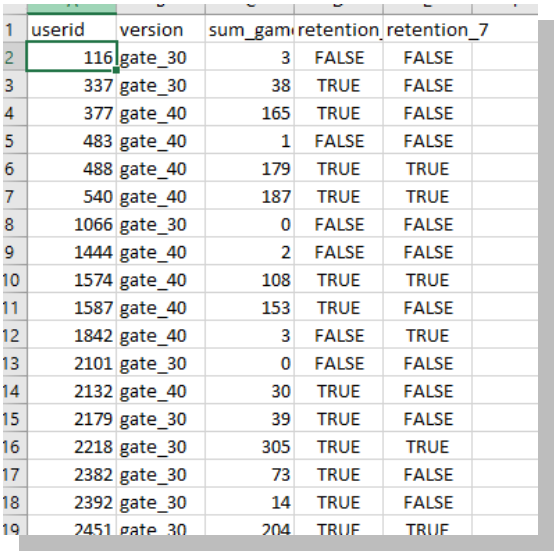
Introduction

A/B Testing merupakan metode eksperimen yang digunakan untuk membandingkan dua versi dari suatu hal, bisa itu website, aplikasi, email, atau iklan. Hal ini, untuk melihat mana yang lebih efektif dalam mencapai tujuan tertentu.

A/B Testing ini penting sekali digunakan dalam perusahaan-perusahaan di dunia, terutama dalam pengambilan keputusan berbasis data. Dengan A/B testing, perusahaan bisa menguji berbagai strategi atau perubahan sebelum diimplementasikan secara luas, sehingga **meminimalkan risiko** dan **memaksimalkan hasil.**

Kali ini, saya akan melakukan eksperimen A/B Testing pada suatu game dari playstore dengan nama **Cookie Cats**. Game tersebut merupakan game puzzle berbasis match-3 yang dikembangkan oleh Tactile Games. Konsepnya mirip dengan **Candy Crush**, di mana pemain harus mencocokkan tiga atau lebih kue dengan warna yang sama untuk menyelesaikan level.

Adapun dataset yang digunakan, berisikan beberapa kolom yang dapat dilihat sebagai berikut.



	userid	version	sum_gam	retention	retention_7
2	116	gate_30	3	FALSE	FALSE
3	337	gate_30	38	TRUE	FALSE
4	377	gate_40	165	TRUE	FALSE
5	483	gate_40	1	FALSE	FALSE
6	488	gate_40	179	TRUE	TRUE
7	540	gate_40	187	TRUE	TRUE
8	1066	gate_30	0	FALSE	FALSE
9	1444	gate_40	2	FALSE	FALSE
10	1574	gate_40	108	TRUE	TRUE
11	1587	gate_40	153	TRUE	FALSE
12	1842	gate_40	3	FALSE	TRUE
13	2101	gate_30	0	FALSE	FALSE
14	2132	gate_40	30	TRUE	FALSE
15	2179	gate_30	39	TRUE	FALSE
16	2218	gate_30	305	TRUE	TRUE
17	2382	gate_30	73	TRUE	FALSE
18	2392	gate_30	14	TRUE	FALSE
19	2451	gate_30	204	TRUE	TRUE

1. **userid**: Nomor unik yang mengidentifikasi setiap pemain.
2. **version**: Apakah pemain dimasukkan ke dalam grup kontrol (gate_30 → gerbang di level 30) atau grup dengan gerbang yang dipindahkan (gate_40 → gerbang di level 40).
3. **sum_gamerounds**: jumlah putaran permainan yang dimainkan pemain selama 14 hari pertama setelah instalasi.
4. **retensi_1**: Apakah pemutar kembali dan bermain 1 hari setelah instalasi?
5. **retensi_7**: Apakah pemutar kembali dan bermain 7 hari setelah pemasangan?

Objective Definition

Sebelum melakukan **A/B Testing**, perlu kita bedah terkait *objective definition* dalam game **Cookie Cats**.

Adapun tujuannya:

Mengukur dampak posisi gate terhadap jumlah ronde yang dimainkan
Menentukan apakah memindahkan gate dari level 30 ke level 40 dapat membuat pemain bermain lebih banyak (dari segi sum_gamerounds) dalam 14 hari pertama.

Hipotesis Formulation

Adapun **formulasi hipotesis** dari tujuan sebelumnya yang telah dirumuskan dapat ditinjau sebagai berikut.

Hipotesis:

Null Hypothesis (H_0)*:*
Tidak ada perbedaan signifikan dalam jumlah ronde permainan (sum_gameround) antara pemain di gate_30 dan gate_40
Alternative Hypothesis (H_1)*:*
Ada perbedaan signifikan dalam jumlah ronde permainan (sum_gameround) antara pemain di gate_30 dan gate_40

Link Google Colab

(My Github Project)

Test Design

Adapun **test design** dari tujuan terkait mengukur dampak posisi gate terhadap jumlah ronde yang dimainkan adalah sebagai berikut.

Group:

Control Group: **Pengguna yang memainkan versi game dengan paywall di level 30 (gate_30)**
Target Group: **Pengguna yang memainkan versi game dengan paywall di level 40 (gate_40)**

Sample Size:

Tujuan: **Menentukan ukuran sampel yang cukup besar untuk mendeteksi perbedaan yang signifikan secara statistik antara kedua grup.**
Menggunakan Perhitungan Sampel Statistik sebagai berikut:

$$\frac{\frac{Z^2 \cdot p(1-p)}{e^2}}{1+\left(\frac{Z^2 \cdot p(1-p)}{e^2 N}\right)}$$

Adapun tahapan perhitungan sampel statistik sebagai berikut:

1. Menentukan proporsi masing-masing gate dimana ketentuan pemain yang bermain > 10 ronde

```
[158] import pandas as pd

      threshold = 10  # Misalnya, pemain yang bermain lebih dari 10 ronde dianggap aktif
      df_30 = df[df["version"] == "gate_30"]
      df_40 = df[df["version"] == "gate_40"]

      # Hitung proporsi pemain yang bermain lebih dari threshold ronde
      p_30 = (df_30["sum_gamerounds"] > threshold).mean()
      p_40 = (df_40["sum_gamerounds"] > threshold).mean()

      print(f"Proporsi pemain gate_30 yang bermain lebih dari {threshold} ronde: {p_30:.2f}")
      print(f"Proporsi pemain gate_40 yang bermain lebih dari {threshold} ronde: {p_40:.2f}")

→ Proporsi pemain gate_30 yang bermain lebih dari 10 ronde: 0.60
  Proporsi pemain gate_40 yang bermain lebih dari 10 ronde: 0.60
```

2. Menghitung proporsi pooled (gabungan) dari gate_30 dan gate_40 untuk menjadi proporsi (p)

```
[158] import pandas as pd

      threshold = 10  # Misalnya, pemain yang bermain lebih dari 10 ronde dianggap aktif
      df_30 = df[df["version"] == "gate_30"]
      df_40 = df[df["version"] == "gate_40"]

      # Hitung proporsi pemain yang bermain lebih dari threshold ronde
      p_30 = (df_30["sum_gamerounds"] > threshold).mean()
      p_40 = (df_40["sum_gamerounds"] > threshold).mean()

      print(f"Proporsi pemain gate_30 yang bermain lebih dari {threshold} ronde: {p_30:.2f}")
      print(f"Proporsi pemain gate_40 yang bermain lebih dari {threshold} ronde: {p_40:.2f}")

→ Proporsi pemain gate_30 yang bermain lebih dari 10 ronde: 0.60
  Proporsi pemain gate_40 yang bermain lebih dari 10 ronde: 0.60
```

```python
import math

def calculate_sample_size(N, p, e, Z=1.96):
    """
    Menghitung ukuran sampel berdasarkan rumus di gambar.

    Parameter:
    N  : Ukuran populasi
    p  : Proporsi populasi (misal 0.5 jika tidak diketahui)
    e  : Margin of error (dalam desimal, misal 0.05 untuk 5%)
    Z  : Skor Z (default 1.96 untuk confidence level 95%)

    Return:
    n  : Ukuran sampel yang dibutuhkan
    """
    numerator = (Z**2 * p * (1 - p)) / (e**2)
    denominator = 1 + ((Z**2 * p * (1 - p)) / (e**2 * N))

    n = numerator / denominator
    return math.ceil(n)

# Contoh penggunaan
N = 90189  # Ukuran populasi
p = p_pooled  # Proporsi populasi
e = 0.05  # Margin of error 5%
Z = 1.96  # Skor Z untuk confidence level 95%

sample_size = calculate_sample_size(N, p, e, Z)
print(f"Ukuran sampel yang dibutuhkan: {sample_size}")
```

```
Ukuran sampel yang dibutuhkan: 367
```

3. Menghitung sampel size dengan rumus yang tertera sebelumnya

Setelah dilakukan perhitungan, didapatkan ukuran sample size untuk dataset Cookies

Cats dari setiap gatenya sebesar **367 data**. Sehingga, setiap group akan diambil **367 data** secara acak.

Randomization:

Tujuan: **Memastikan bahwa pembagian pengguna ke dalam control dan target group dilakukan secara acak untuk menghindari bias.**
Jadi data akan diambil sejumlah n sample size di setiap groupnya secara acak agak tidak bias.
Metode yang digunakan: Sampling Acak Random

```python
import pandas as pd

# Tentukan jumlah sampel per grup
n_sample = sample_size

# Ambil sampel acak dari setiap grup berdasarkan 'gate_level'
sample_df = df.groupby("version").sample(n=n_sample, random_state=42)

# Tampilkan hasil sampel
sample_df.head()
```

	userid	version	sum_gamerounds	retention_1	retention_7
58980	6531033	gate_30	35	False	False
72247	8003009	gate_30	4	False	False
50215	5566807	gate_30	23	True	False
609	64235	gate_30	1	False	False
56038	6210551	gate_30	7	False	False

(Gambar pengambilan sampel secara acak)

Duration:

Tujuan: **Menentukan durasi tes yang cukup untuk mengumpulkan data yang memadai, sehingga bisa mendeteksi perubahan dalam jumlah ronde yang dimainkan.**
Uji dijalankan minimal selama **14 hari**, sesuai dengan durasi pengamatan jumlah ronde dalam dua minggu pertama.
Pertimbangkan untuk memperpanjang durasi jika: - Jumlah sampel belum mencukupi sesuai hasil perhitungan sampel - Ada fluktuasi aktivitas pemain (misalnya, saat liburan atau event khusus). **Pastikan tes mencakup variasi hari dalam seminggu untuk menghindari bias dari pola bermain yang berbeda di hari kerja vs. akhir pekan.**

Data Collection

Adapun **data collection** dari tujuan terkait mengukur dampak posisi gate terhadap jumlah ronde yang dimainkan adalah sebagai berikut.

Dataset Cookies Cats:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90189 entries, 0 to 90188
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   userid          90189 non-null  int64
 1   version         90189 non-null  object
 2   sum_gamerounds  90189 non-null  int64
 3   retention_1     90189 non-null  bool
 4   retention_7     90189 non-null  bool
dtypes: bool(2), int64(2), object(1)
memory usage: 2.2+ MB
```

	userid	version	sum_gamerounds	retention_1	retention_7
0	116	gate_30	3	False	False
1	337	gate_30	38	True	False
2	377	gate_40	165	True	False
3	483	gate_40	1	False	False
4	488	gate_40	179	True	True
...	...	...	...	...	...
95	9443	gate_40	3	False	False
96	9589	gate_30	8	False	False
97	9699	gate_40	8	False	False
98	9828	gate_40	1	False	False
99	9831	gate_40	3	False	False

Berdasarkan hasil output di atas, adapun dataset **Cookies Cats** didapatkan rincian nama variabel sebagai berikut:

- **userid**: Nomor unik yang mengidentifikasi setiap pemain. (Tipe datanya Numerik)
- **version**: Apakah pemain dimasukkan ke dalam grup kontrol (gate_30 → gerbang di level 30) atau grup dengan gerbang yang dipindahkan (gate_40 → gerbang di level 40). (Tipe datanya string)
- **sum_gamerounds**: jumlah putaran permainan yang dimainkan pemain selama 14 hari pertama setelah instalasi. (Tipe datanya) Numerik)

- **retensi_1**: Apakah pemutar kembali dan bermain 1 hari setelah instalasi?. (Tipe datanya Boolean karena True/False)
- **retensi_7**: Apakah pemutar kembali dan bermain 7 hari setelah pemasangan?. (Tipe datanya Boolean karena True/False)

Selain itu, Identifikasi metrik yang ingin diukur dari dataset Cookies Cats untuk A/B Testing:

- **User Engagement**: jumlah ronde permainan dalam **14 hari pertama**.

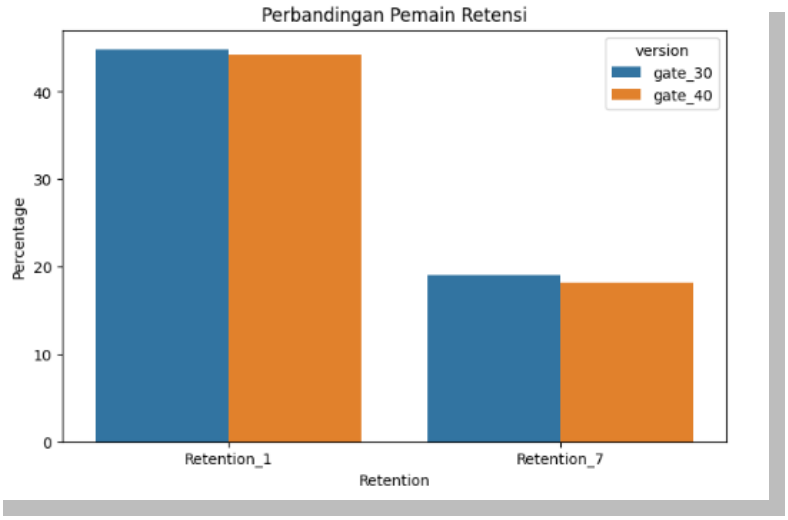
User Engagement :

	count	mean	std	min	25%	50%	75%	max
version								
gate_30	44700.0	52.456264	256.716423	0.0	5.0	17.0	50.0	49854.0
gate_40	45489.0	51.298776	103.294416	0.0	5.0	16.0	52.0	2640.0

Insights:

Rata-rata jumlah ronde permainan di gate_30 sedikit lebih tinggi daripada gate_40.
Variasi di gate_30 sangat besar, sementara gate_40 lebih stabil.
Gate_40 mungkin lebih mengontrol keterlibatan pemain, sedangkan gate_30 menghasilkan lebih banyak outlier dengan jumlah permainan yang ekstrem.

- **User Retention**: Persentase pemain yang kembali bermain setelah **1 dan 7 hari**.




```
Retention Day 1:
version
gate_30    44.818792
gate_40    44.228275
Name: retention_1, dtype: float64
Retention Day 7:
version
gate_30    19.020134
gate_40    18.200004
Name: retention_7, dtype: float64
```

Insights:

Gate_30 memiliki retensi lebih baik di hari ke-1 dan ke-7, tetapi selisihnya sangat kecil dibandingkan gate_40.

Tidak ada dampak besar dalam retensi akibat pemindahan gate dari level 30 ke level 40.

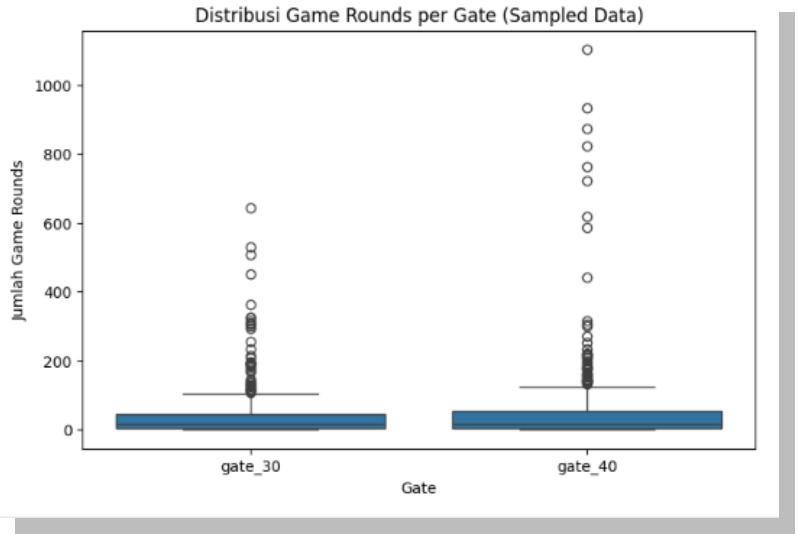
Data Analysis

Adapun **data analysis** dimana untuk menjawab tujuan dan hipotesis yang diduga yang sudah dirumuskan sebelumnya. Dalam analisis data ini melakukan pengujian hipotesis menggunakan Pengujian 2 populasi secara paired/tidak.

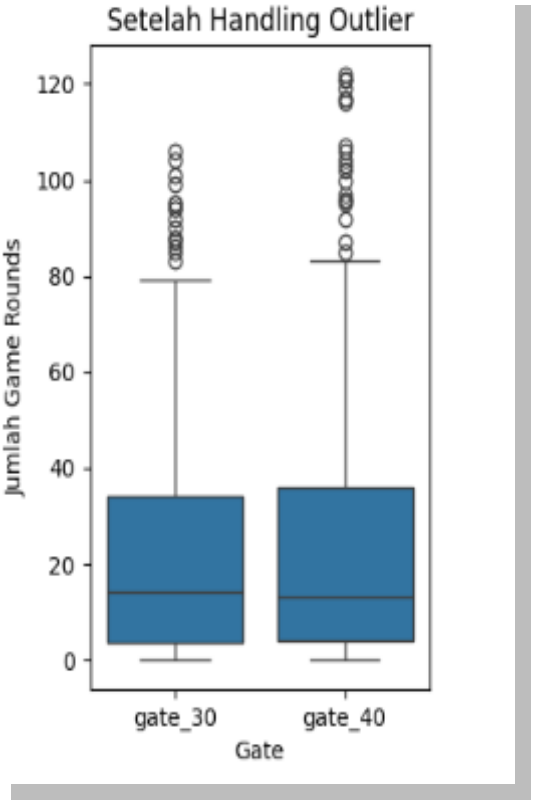
Setelah pengambilan sampel secara acak sebelumnya, maka ada tahapan yang perlu dilakukan:

1. **Mengecek** Outlier dari data set gate_30 dan gate_40

Menggunakan **Boxplot**, didapatkan bahwa gate_30 dan gate_40 terdapat indikasi data ekstrem / outlier di atas nilai maksimum.



2. Melakukan Handling outlier dengan pendekatan Interquartile **(IQR)**



Setelah dilakukan Handling, terlihat bahwa masih ada data ekstrem namun kemungkinna besar datanya mendekati nilai maksimum. Secara visual datanya berhimpitan, sehingga dianggap wajar dan bukan outlier. Intinya:

- Data memang memiliki distirbusi yang skewed.

- Data ekstrem masih dalam pola wajar karena nilainya make sense/ mendekati nilai yang lain.

- Bukan noise atau kesalahan data.

Nah, *kemudian* nantinya data akan dilakukan tahapan mengenai **Data Tanpa Outlier Vs Data Sama Outlier** (Perbandingan) → Tahapannya

1. Pengecekan asumsi normalitas menggunakan **Kolmogorov Smirnov. (Outlier Vs Tanpa Outlier Dataset)**
   a. Jika data berdistribusi normal maka menggunakan Pengujian Hipotesis 2 Populasi Parametrik (**T-Test atau Paired**).
   b. Jika data tidak berdistribusi normal maka menggunakan Pengujian Hipotesis 2 Populasi Non Parametrik (**Mann Whitney** atau **Wilcoxon Rank Test**).
2. Pengujian hipotesis statistic parametrik/non-parametrik **(Outlier Vs Tanpa Outlier Dataset)**
3. Kesimpulan hasil pengujian hipotesis statistic parametrik/non-parametrik (Outlier Vs Tanpa Outlier Dataset.

Data Analysis (Result)

1. Pengecekan asumsi normalitas menggunakan **Kolmogorov Smirnov. (Outlier Vs Tanpa Outlier Dataset)**

Hipotesis:

- **Outlier**

H_0: Data Outlier berdistribusi normal
H_1: Data Outlier tidak berdistribusi normal

- **Tanpa Outlier**

H_0: Data Tanpa Outlier berdistribusi normal
H_1: Data Tanpa Outlier tidak berdistribusi normal

Taraf Signifikansi (alpha) = 5%
Daerah Penolakan = Tolak H_0, jika nilai *p-value* < 5%
Hasil →

```
[168] from scipy.stats import kstest

[169] #Data Tanpa Outlier
      stat, p = kstest(df_clean['sum_gamerounds'], 'norm')
      print("p-value =", p)

      p-value = 0.0

[170] #Data Sama Outlier
      stat, p = kstest(sample_df['sum_gamerounds'], 'norm')
      print("p-value =", p)

      p-value = 0.0
```

Interpretasi:

Outlier: Tolak H_0, karena nilai *p-value* < 5%. Sehingga keputusannya Data Outlier tidak berdistribusi normal.

Tanpa Outlier: Tolak H_0, karena nilai *p-value* < 5%. Sehingga keputusannya Data Tanpa Outlier tidak berdistribusi normal.

Nah, karena Data semuanya tidak berdistribusi normal maka menggunakan Pengujian Hipotesis **Statistik Non-parametrik**. Jenis yang digunakan adalah **Mann-Whitney** karena dataset nya merupakan independen atau tidak menujukkan observasi sebelum dan sesudah (populasinya berbeda keduanya antara **Gate 30 dengan Gate 40**)

---

2. Pengujian hipotesis statistic parametrik/non-parametrik (Outlier Vs Tanpa Outlier Dataset)

Hipotesis:

- **Outlier**

H_0: Tidak ada perbedaan signifikan dalam jumlah ronde permainan antara gate 30 dan gate 40

H_1: Terdapat perbedaan signifikan dalam jumlah ronde permainan antara gate 30 dan gate 40

- **Tanpa Outlier**

H_0: Tidak ada perbedaan signifikan dalam jumlah ronde permainan antara gate 30 dan gate 40

H_1: Terdapat perbedaan signifikan dalam jumlah ronde permainan antara gate 30 dan gate 40

Taraf Signifikansi (alpha) = 5%

Daerah Penolakan = Tolak H_0, jika nilai *p-value* < 5%

Hasil →

- Outlier

```
from scipy.stats import mannwhitneyu

# Misal data sudah dalam bentuk list
sum_gameround_gate_30 = sample_df.loc[sample_df["version"] == "gate_30", "sum_gamerounds"] #gate_30
sum_gameround_gate_40 = sample_df.loc[sample_df["version"] == "gate_40", "sum_gamerounds"] #gate_40

stat, p = mannwhitneyu(sum_gameround_gate_30, sum_gameround_gate_40, alternative='two-sided')

print("Mann-Whitney U Test p-value:", p)

if p > 0.05:
    print("Tidak ada perbedaan signifikan dalam jumlah ronde permainan antara gate_30 dan gate_40")
else:
    print("Ada perbedaan signifikan dalam jumlah ronde permainan antara gate_30 dan gate_40")
```

```
Mann-Whitney U Test p-value: 0.7689121002513506
Tidak ada perbedaan signifikan dalam jumlah ronde permainan antara gate_30 dan gate_40
```

- Tanpa Outlier

```
from scipy.stats import mannwhitneyu

# Misal data sudah dalam bentuk list
sum_gameround_gate_30 = df_clean.loc[df_clean["version"] == "gate_30", "sum_gamerounds"] #gate_30
sum_gameround_gate_40 = df_clean.loc[df_clean["version"] == "gate_40", "sum_gamerounds"] #gate_40

stat, p = mannwhitneyu(sum_gameround_gate_30, sum_gameround_gate_40, alternative='two-sided')

print("Mann-Whitney U Test p-value:", p)

if p > 0.05:
    print("Tidak ada perbedaan signifikan dalam jumlah ronde permainan antara gate_30 dan gate_40")
else:
    print("Ada perbedaan signifikan dalam jumlah ronde permainan antara gate_30 dan gate_40")
```

```
Mann-Whitney U Test p-value: 0.8454702076397356
Tidak ada perbedaan signifikan dalam jumlah ronde permainan antara gate_30 dan gate_40
```

Interpretasi:

Outlier: Tolak H_0, karena nilai *p-value* < 5%. Sehingga keputusannya Tidak ada perbedaan signifikan dalam jumlah ronde permainan antara gate 30 dan gate 40.

Tanpa Outlier: Tolak H_0, karena nilai *p-value* < 5%. Sehingga keputusannya Tidak ada perbedaan signifikan dalam jumlah ronde permainan antara gate 30 dan gate 40.

Dengan demikian, penerapan **gate 30** dan **gate 40** tidak memiliki dampak yang berbeda terhadap jumlah ronde permainan pemain (Fitur baru tidak memberikan dampak apa-apa).

Interpretation/Conclution for company

Kesimpulan dalam Bisnis

1. Perubahan dari gate_30 ke gate_40 tidak memengaruhi jumlah ronde permainan secara signifikan → **Jika tujuan eksperimen adalah meningkatkan engagement, maka perubahan ini tidak memberikan dampak yang diharapkan**.
2. Tidak ada risiko besar dalam mempertahankan versi mana pun → Karena tidak ada perbedaan signifikan, **perusahaan bisa memilih untuk mempertahankan gate_30 atau gate_40 berdasarkan faktor lain** (misalnya, kemudahan implementasi atau biaya operasional).
3. Mungkin ada faktor lain yang lebih berpengaruh terhadap jumlah ronde permainan → **Fitur lain dalam game, mekanisme reward, atau pengalaman pengguna mungkin lebih berperan dibanding sekadar perbedaan gate**.

Saran/Rekomendasi

1. Mengevaluasi Faktor lain yang bisa memengaruhi jumlah Ronde, mungkin menunjukkan perubahan gate tidak cukup.
2. Bisa melakukan segmentasi data, terkait pola datanya seperti, sehingga bisa diketahui efek perubahannya. Salah satu contoh: **efek perubahan gate hanya terasa pada kelompok pemain tertentu (pemain lama vs pemain baru atau pemain remaja vs pemain anak-anak).**
3. Bisa melakukan pengujian lain yang lebih **spesifik**, dari sisi reward milestone, dari sisi misi atau tantangan, atau dari fitur gamenya.
4. Perusahaan sebaiknya **mempertimbangkan faktor lain** sebelum memutuskan apakah ingin mempertahankan atau mengubah sistem gate.


-Terima Kasih-