



**Final Project**

# **DTS FGA x Binar Academy Data Science**

**Muhammad Iqbal Yoshanda**



A group of people are seated in a room, possibly a lecture hall or meeting space. In the background, a large, stylized sign reads 'BINAR'. The image is overlaid with a dark blue filter.

“

**Information is the oil of the 21<sup>st</sup> century, and analytics is the combustion engine.**

- Peter Sondergaard

# Table of Contents



## Challenge Chapter 1

- Melakukan Query di Google BigQuery
- Merancang Dashboard



## Challenge Chapter 2

- Business Understanding
  - Exploratory Data Analysis
  - Preprocessing
- Modeling
  - Evaluation
  - Prediction



## Kesimpulan

- Kesimpulan
- Ringkasan Insight
- Rekomendasi



A group of people are sitting in a circle in a room. In the background, large letters spell out 'BINAR' on the wall. The image is dark and has a blue tint.

## **Challenge Chapter 1**

# Project Overview – Analisis Covid-19 Indonesia



Pada studi kasus ini pada tahap pertama akan dilakukan analisa terhadap data kasus Covid-19 yang terjadi di Indonesia, dimana dilakukan pengolahan data dengan query pada Google BigQuery. Pada tahap yang kedua akan dilakukan perancangan dashboard Covid-19 Indonesia dengan menggunakan Looker Studio.



# Melakukan Query di BigQuery



Google  
Big Query

## Jumlah total kasus Covid-19 aktif yang baru di setiap provinsi lalu diurutkan berdasarkan jumlah kasus yang paling besar

### SQL Query

```
SELECT
Province,
SUM(New_Active_Cases) AS Total_Kasus_Aktif_Baru
FROM `atomic-router-414914.challenge_chapter1.kasus_covid19_indonesia`
WHERE Location_Level = 'Province'
GROUP BY 1
ORDER BY 2 DESC;
```

Row	Province	Total_Kasus_Aktif_Baru
1	Jawa Barat	13496
2	DKI Jakarta	10922
3	Banten	2558
4	Jawa Tengah	1423
5	Jawa Timur	1136
6	Daerah Istimewa Yogyakarta	669
7	Sumatera Utara	664
8	Sulawesi Utara	565
9	Bali	474
10	Sumatera Selatan	313

### Insight

Top 3 provinsi dengan jumlah kasus Covid-19 aktif yang paling tinggi:

1. Jawa Barat
2. DKI Jakarta
3. Banten



## Mengambil 2 location iso code yang memiliki jumlah total kematian karena Covid-19 paling sedikit

### SQL Query

```
SELECT
Location_ISO_Code,
Location,
SUM(Total_Deaths) AS Total_Kematian
FROM `atomic-router-
414914.challenge_chapter1.kasus_covid19_indonesia`
GROUP BY 1, 2
ORDER BY 3 ASC
LIMIT 2;
```

Row	Location_ISO_Code	Location	Total_Kematian
1	ID-MA	Maluku	147196
2	ID-MU	Maluku Utara	167511

### Insight

Maluku (ID-MA) dan Maluku Utara (ID-MU) menjadi provinsi dengan total kematian karena Covid-19 paling sedikit diantara provinsi yang lain.





# Data tentang tanggal-tanggal ketika rate kasus recovered di Indonesia paling tinggi

## SQL Query

```
WITH recovered AS(
  SELECT
    Date,
    FORMAT_DATE('%A', Date) AS Day,
    FORMAT_DATE('%B', Date) AS Month,
    FORMAT_DATE('%Y', Date) AS Year,
    ROUND(Case_Recovered_Rate, 2) AS Case_Recovered_Rate,
    ROW_NUMBER() OVER (PARTITION BY FORMAT_DATE('%B', Date), FORMAT_DATE('%Y', Date) ORDER BY Case_Recovered_Rate DESC) AS Partition_Number
  FROM `atomic-router-414914.challenge_chapter1.kasus_covid19_indonesia`
)

SELECT
  Date,
  Day,
  Month,
  Year,
  Case_Recovered_Rate AS Max_Case_Recovered_Rate
FROM recovered
WHERE Partition_Number = 1
ORDER BY 1 ASC;
```



## Data tentang tanggal-tanggal ketika rate kasus recovered di Indonesia paling tinggi

Row	Date ▾	Day ▾	Month ▾	Year ▾	Max_Case_Recovere
1	2020-03-06	Friday	March	2020	111.0
2	2020-04-01	Wednesday	April	2020	5.86
3	2020-05-02	Saturday	May	2020	0.97
4	2020-06-03	Wednesday	June	2020	0.95
5	2020-07-22	Wednesday	July	2020	0.99
6	2020-08-12	Wednesday	August	2020	0.96
7	2020-09-05	Saturday	September	2020	0.95
8	2020-10-22	Thursday	October	2020	0.95
9	2020-11-23	Monday	November	2020	0.97
10	2020-12-01	Tuesday	December	2020	0.95

### Insight

Pada tanggal 6 Maret 2020, rate kasus recovered mencapai 111%. Hal ini menjadikannya tanggal dengan rate kasus recovered tertinggi selama periode Covid-19 yang terjadi di Indonesia.



## Total case fatality rate dan case recovered rate dari masing-masing location iso code yang diurutkan dari data yang paling rendah

### SQL Query

```
SELECT
Location_ISO_Code,
Location,
SUM(Case_Fatality_Rate) AS Total_Case_Fatality_Rate,
SUM(Case_Recovered_Rate) AS Total_Case_Recovered_Rate
FROM `atomic-router-
414914.challenge_chapter1.kasus_covid19_indonesia`
GROUP BY 1, 2
ORDER BY 3, 4 ASC
```

Row	Location_ISO_Code	Location	Total_Case_Fatality_Rate	Total_Case_Recovered_Rate
1	ID-KU	Kalimantan Utara	14.285000000000021	733.72659999999894
2	ID-NT	Nusa Tenggara Ti...	15.934500000000002	700.82079999999894
3	ID-PA	Papua	16.895300000000013	608.23260000000084
4	ID-JA	Jambi	17.326799999999977	760.52920000000131
5	ID-SG	Sulawesi Tenggara	19.668699999999919	741.66440000000159
6	ID-KB	Kalimantan Barat	20.560999999999932	771.57379999999887
7	ID-SR	Sulawesi Barat	21.755600000000072	732.87229999999954
8	ID-SN	Sulawesi Selatan	22.457400000000142	775.29740000000061
9	ID-SB	Sumatera Barat	24.010300000000047	754.2531
10	ID-PB	Papua Barat	24.334100000000088	757.19869999999998

### Insight

Top 3 provinsi dengan total case fatality rate paling rendah:

1. Kalimantan Utara (ID-KU)
2. Nusa Tenggara Timur (ID-NT)
3. Papua (ID-PA)



## Data tentang tanggal-tanggal saat total kasus Covid-19 mulai menyentuh angka 30.000-an

### SQL Query

```
SELECT
Date,
SUM(Total_Cases) AS Total_Kasus,
FROM `atomic-router-
414914.challenge_chapter1.kasus_covid19_indonesia`
WHERE Total_Cases >= 30000
GROUP BY 1
ORDER BY 2 ASC
```

Row	Date	Total_Kasus
1	2020-06-06	30514
2	2020-06-07	31186
3	2020-06-08	32033
4	2020-06-09	33075
5	2020-06-10	34316
6	2020-06-11	35295
7	2020-06-12	36406
8	2020-06-13	37420
9	2020-06-14	38277
10	2020-06-15	39294

### Insight

Total kasus Covid-19 mulai menyentuh angka 30.000-an pada tanggal 6 Juni 2020.



## Jumlah data yang tercatat ketika kasus Covid-19 lebih dari atau sama dengan 30.000

### SQL Query

```
SELECT  
COUNT(*) AS Jumlah_Data  
FROM `atomic-router-  
414914.challenge_chapter1.kasus_covid19_indonesia`  
WHERE Total_Cases >= 30000;
```

Row	Jumlah_Data
1	14399

### Insight

Terdapat 14.399 data yang tercatat untuk total kasus ketika kasus Covid-19 lebih dari atau sama dengan 30.000.



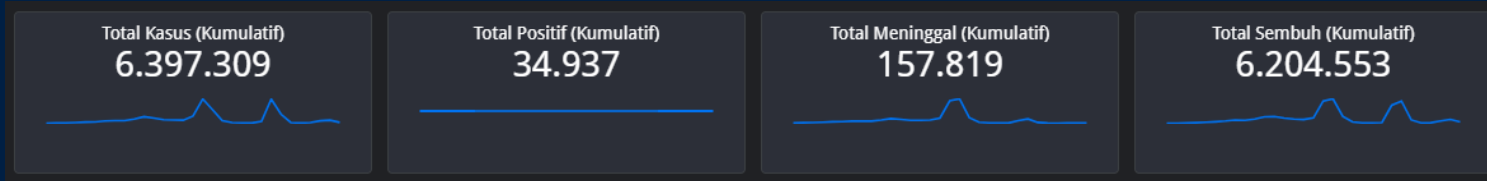
# Merancang Dashboard



Looker

PART OF Google Cloud

# Scorecards



Scorecards ini memberikan informasi cepat tentang kasus Covid-19 yang terjadi di Indonesia, dimana menampilkan total kasus (total keseluruhan kasus yang terjadi), total positif (total kasus positif yang masih aktif), total meninggal (total kasus meninggal), dan total sembuh (total sembuh dari Covid-19). Dengan ini, scorecards dapat memberikan gambaran singkat mengenai situasi tentang Covid-19 yang terjadi di Indonesia serta perkembangannya dari waktu ke waktu.



# Bar Charts

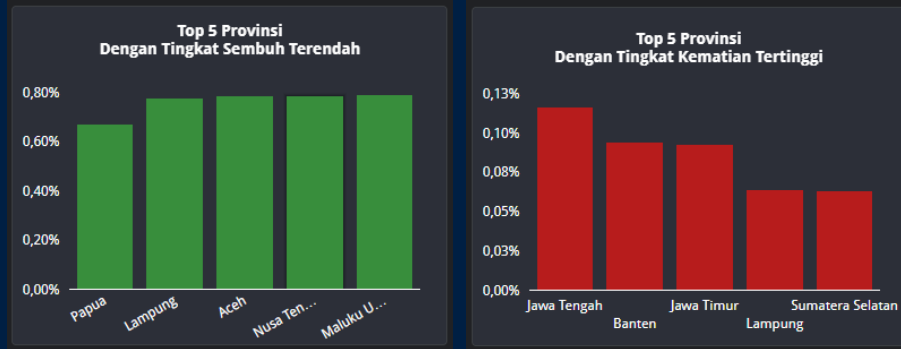


Bar Charts menggunakan data provinsi menampilkan perbandingan visual antara 5 provinsi yang diantaranya provinsi dengan kasus positif tertinggi, provinsi dengan kasus meninggal tertinggi, dan provinsi dengan kasus sembuh tertinggi. Adanya bar charts ini dapat membantu dengan mudah melihat dampak dari Covid-19 yang terjadi di berbagai provinsi di Indonesia. Dengan adanya bar charts ini, pengguna tidak hanya dapat melihat dengan jelas provinsi mana yang memiliki jumlah kasus paling tinggi, tetapi juga dapat memahami situasi pandemi secara lebih luas dan lebih mendalam sehingga pengguna dapat mengidentifikasi dengan mudah pola dan fokus pada wilayah-wilayah yang membutuhkan perhatian khusus.





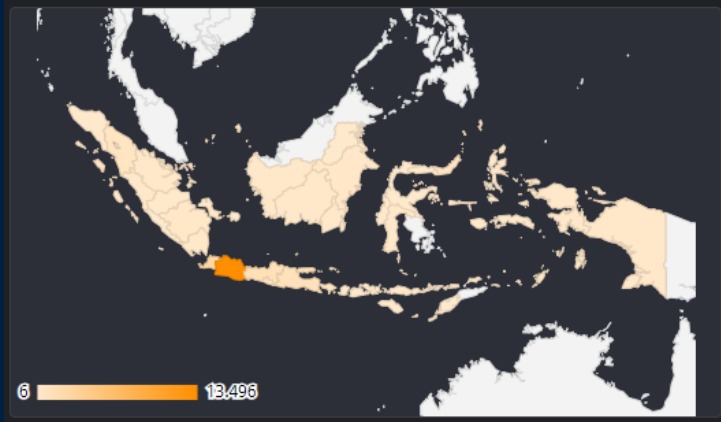
# Bar Charts



Bar Charts menggunakan data provinsi juga menampilkan perbandingan visual antara 5 provinsi yang diantaranya provinsi dengan tingkat kesembuhan terendah, dan provinsi dengan tingkat kematian tertinggi. Dengan adanya bar charts ini, pengguna juga dapat memahami situasi pandemi secara lebih luas dan lebih mendalam sehingga pengguna juga dapat mengidentifikasi dengan mudah pola dan fokus pada wilayah-wilayah yang membutuhkan perhatian khusus.



# Geo Charts



Geo Charts menggunakan data provinsi menampilkan persebaran total kasus positif Covid-19 di seluruh wilayah Indonesia. Dengan adanya geo charts ini, pengguna dapat dengan mudah melihat keadaan di setiap provinsi dan memahami sebaran kasus positif di seluruh Indonesia. Ini sangat penting untuk membantu pengguna, dan stakeholders dalam memahami dan merespons pandemi secara efektif.



# Tables

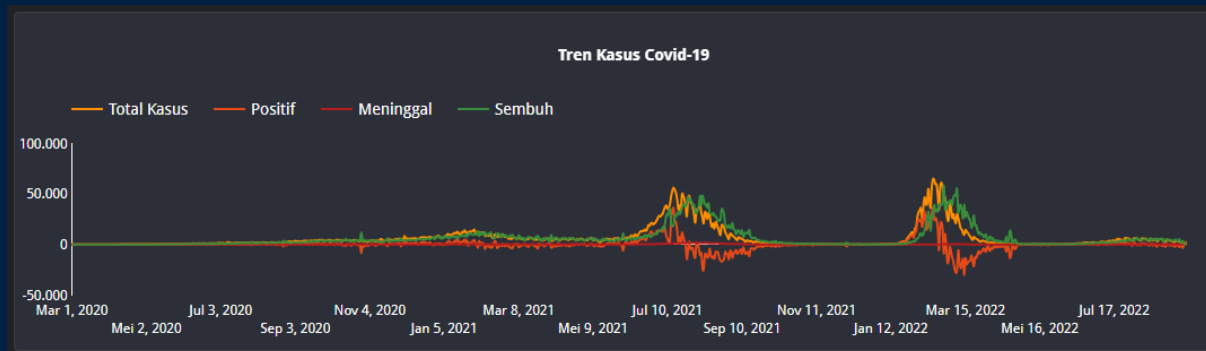
	Provinsi	Total Kasus ▾	Positif	Meninggal	Tingkat Kematian	Sembuh	Tingkat Kesembuhan	P
1.	DKI Jakarta	1,412.474	10.922	15.493	0,04%	1.386.059	0,87%	10
2.	Jawa Barat	1,173.731	13.496	15.937	0,03%	1.144.298	1,06%	45
3.	Jawa Tengah	636.409	1.423	33.480	0,12%	601.506	0,82%	36
4.	Jawa Timur	601.534	1.136	31.732	0,09%	568.666	0,99%	40
5.	Banten	333.875	2.558	2.945	0,09%	328.372	1,91%	10
6.	Daerah Istimewa Yogyakarta	224.307	669	5.928	0,04%	217.710	0,83%	3
7.	Kalimantan Timur	209.017	272	5.726	0,03%	203.019	0,83%	3
8.	Bali	166.831	474	4.731	0,05%	161.626	0,81%	4

1 - 34 / 34 < >

Tables dengan heatmap membantu pengguna untuk melihat lebih detail perbandingan total kasus, kasus positif, kasus meninggal, tingkat kematian, kasus sembuh, tingkat kesembuhan dengan populasi di setiap provinsi untuk memberikan informasi lanjutan terkait konteks demografis.



# Line Charts



Line Charts digunakan untuk menampilkan tren kasus Covid-19 di Indonesia berdasarkan total kasus, total kasus positif, total kasus meninggal, dan total kasus sembuh. Ini akan memudahkan pengguna untuk mengetahui perkembangan kasus Covid-19 dari waktu ke waktu.



# Filter

Pilih rentang tanggal

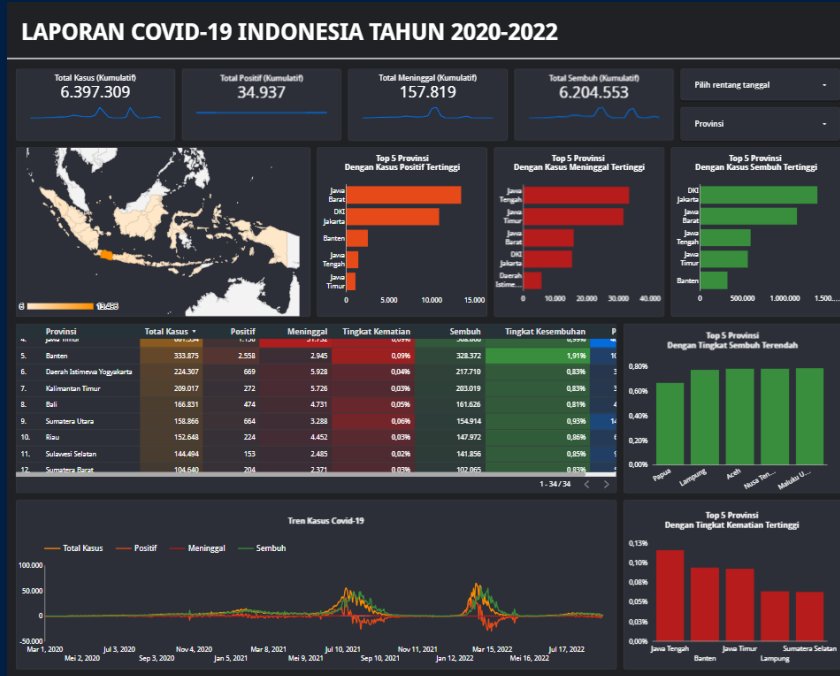
Filter Rentang Tanggal memungkinkan pengguna untuk melakukan analisis berdasarkan rentang tanggal tertentu, membantu mengidentifikasi tren seiring waktu.

Provinsi

Filter Provinsi memungkinkan pengguna untuk melakukan analisis lebih detail berdasarkan provinsi tertentu, membantu memahami situasi kasus setempat yang lebih detail.



# Preview Dashboard



Link Dashboard: [Dashboard Covid-19 Indonesia](#)



A group of students are sitting in a circle in a classroom. In the background, the word "BIMAR" is written in large, stylized letters on the wall. The students are engaged in a discussion or activity. The image has a purple overlay.

## Challenge Chapter 2

## Project Overview – Prediksi Customer Churn

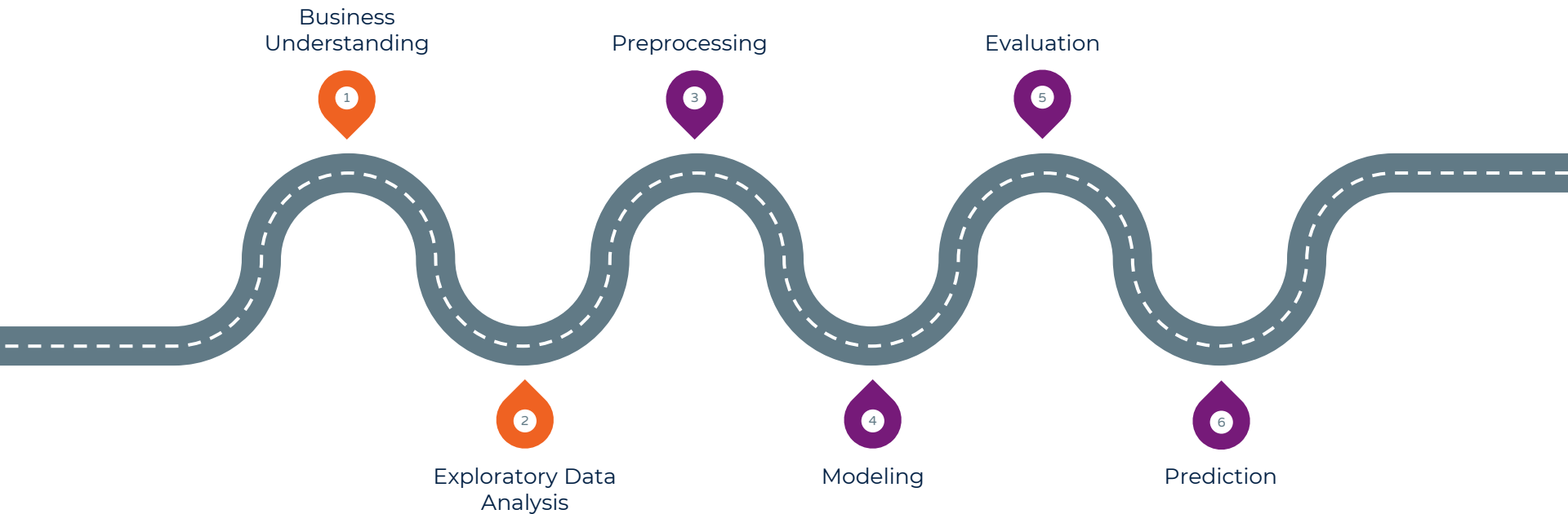
Pada studi kasus ini akan diberikan data terkait customer churn, dimana dari data tersebut akan dilakukan analisa untuk mendapatkan insight serta memprediksi customer churn menggunakan machine learning.

Link Google Colab: [FGA DS Challenge 2](#) 





# Roadmap



# Business Understanding

## Problem Statement

Di era digital yang dipenuhi dengan persaingan yang ketat di industri telekomunikasi, perusahaan-perusahaan sering kali berjuang keras untuk mempertahankan basis customer mereka. Salah satu tantangan utama yang dihadapi oleh penyedia layanan telekomunikasi adalah tingkat churn customer yang tinggi. Churn customer, yang merujuk pada fenomena dimana customer beralih dari satu penyedia layanan ke penyedia lain, dapat mengakibatkan kerugian finansial dan mempengaruhi reputasi perusahaan. Prediksi churn customer menjadi krusial bagi perusahaan telekomunikasi untuk dapat mengantisipasi dan mengurangi kehilangan customer. Oleh karena itu, diperlukan pengembangan Machine Learning yang mampu memprediksi churn customer dengan tingkat akurasi yang optimal, sehingga memungkinkan perusahaan untuk mengambil tindakan proaktif dalam mempertahankan customer dan meningkatkan retensi customer secara efektif.

## Objective

Membangun model klasifikasi menggunakan machine learning yang dapat mengenali dan memprediksi customer yang berpotensi untuk melakukan churn (berhenti berlangganan/beralih) dari layanan telekomunikasi. Dengan demikian perusahaan dapat melakukan pengambilan keputusan yang proaktif dalam mempertahankan customer dan meningkatkan retensi customer secara efektif.



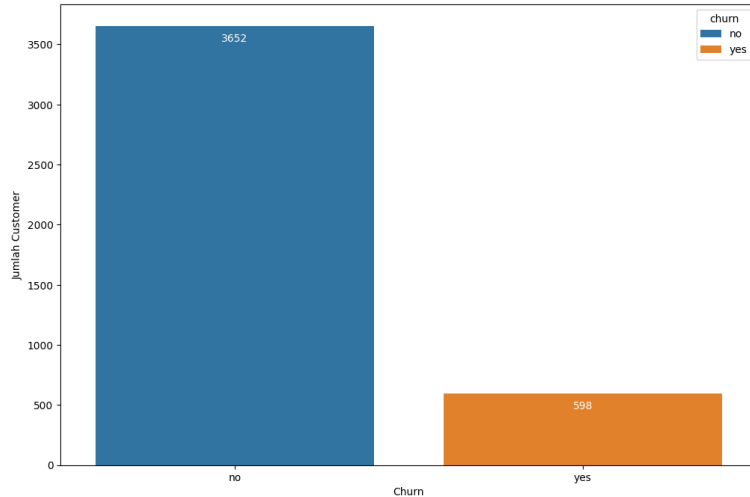
A group of people are seated in a circle in a room that appears to be a classroom or a meeting space. In the background, the word "BINAR" is written in large, stylized letters on the wall. The scene is overlaid with a semi-transparent purple filter.

## **Exploratory Data Analysis**

# Berapa total jumlah customer yang churn?

## Perbandingan Customer Churn dan Retention

Terdapat 598 (14.07%) Customer melakukan churn dan 3.652 (85.93%) Customer tetap berlangganan



## Insight

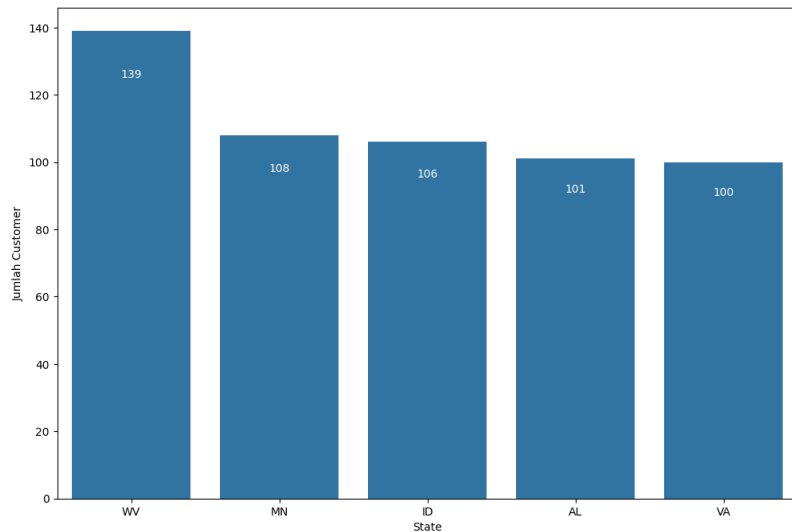
Dari total 4.250 customer terdapat lebih dari 14% (598) customer melakukan churn atau berhenti berlangganan.



# Negara bagian mana yang memiliki jumlah customer terbanyak?

## Top 5 Negara Bagian dengan Jumlah Customer Terbanyak

Customer terbanyak berasal dari negara bagian West Virginia dengan total 139 Customer



## Insight

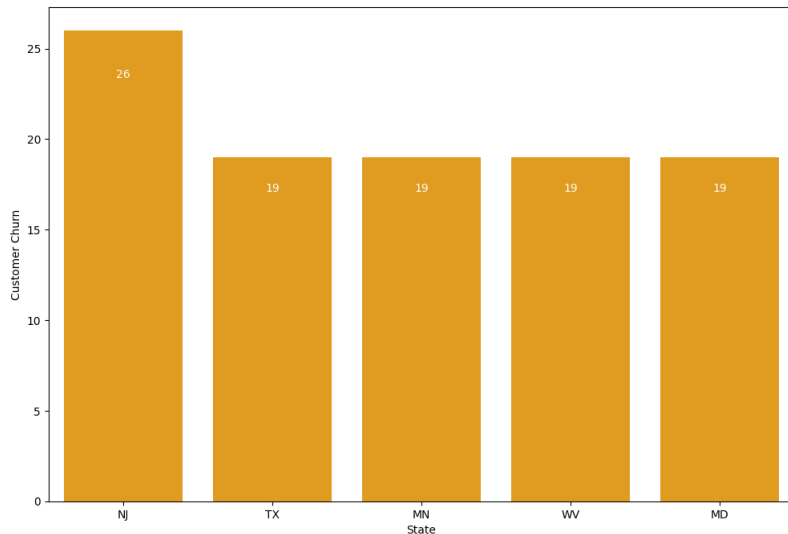
Negara bagian West Virginia merupakan negara bagian dengan jumlah customer terbanyak dengan total jumlah 139 customer.



# Jumlah customer churn terbanyak berasal dari negara bagian mana?

## Top 5 Negara Bagian dengan Customer Churn Terbanyak

Jumlah Customer melakukan Churn terbanyak berasal dari negara bagian New Jersey dengan total 26 Customer



## Insight

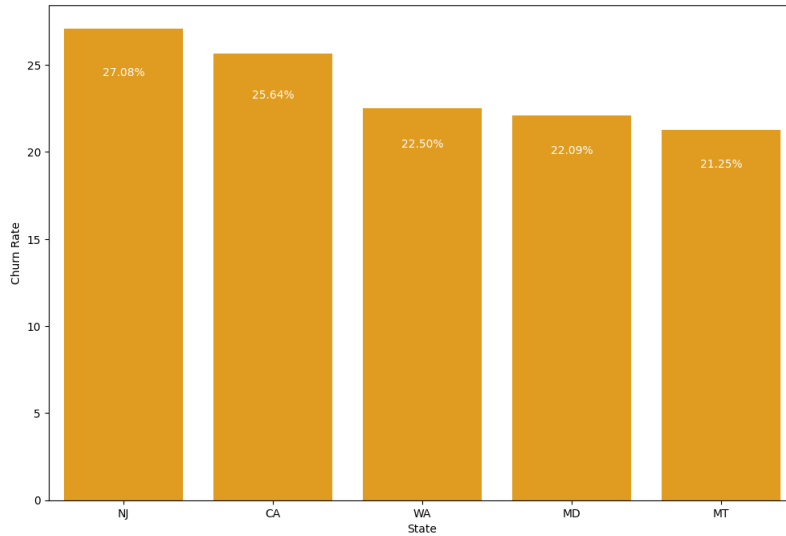
Negara bagian New Jersey merupakan negara bagian dengan jumlah customer churn (berhenti berlangganan) terbanyak dengan jumlah total 26 customer.



# Negara bagian mana yang memiliki tingkat churn tertinggi?

## Top 5 Negara Bagian dengan Tingkat Churn Tertinggi

Negara bagian New Jersey menjadi negara bagian dengan tingkat Churn tertinggi (27.08%)



## Insight

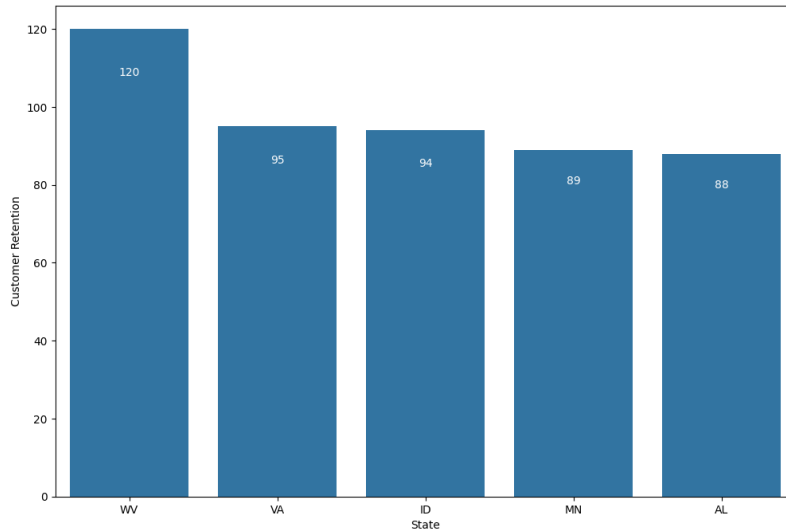
Negara bagian New Jersey selain menjadi negara bagian dengan jumlah customer churn terbanyak juga menjadi negara bagian dengan tingkat churn tertinggi.



# Negara bagian mana yang memiliki jumlah retensi customer terbanyak?

## Top 5 Negara Bagian dengan Retensi Customer Terbanyak

Jumlah Retensi Customer terbanyak berasal dari negara bagian West Virginia dengan total 120 Customer



## Insight

Negara bagian West Virginia merupakan negara bagian dengan jumlah retensi customer (masih bertahan untuk berlangganan) terbanyak dengan jumlah total 120 customer.

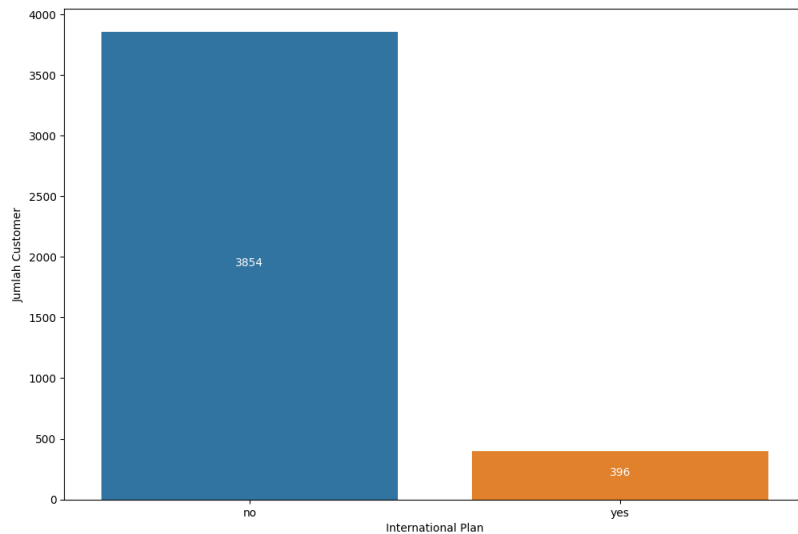




# Berapa jumlah customer yang berlangganan paket internasional?

## Jumlah Customer Berdasarkan Paket Internasional

Customer yang berlangganan paket internasional hanya berjumlah 396 (9.31%) Customer



## Insight

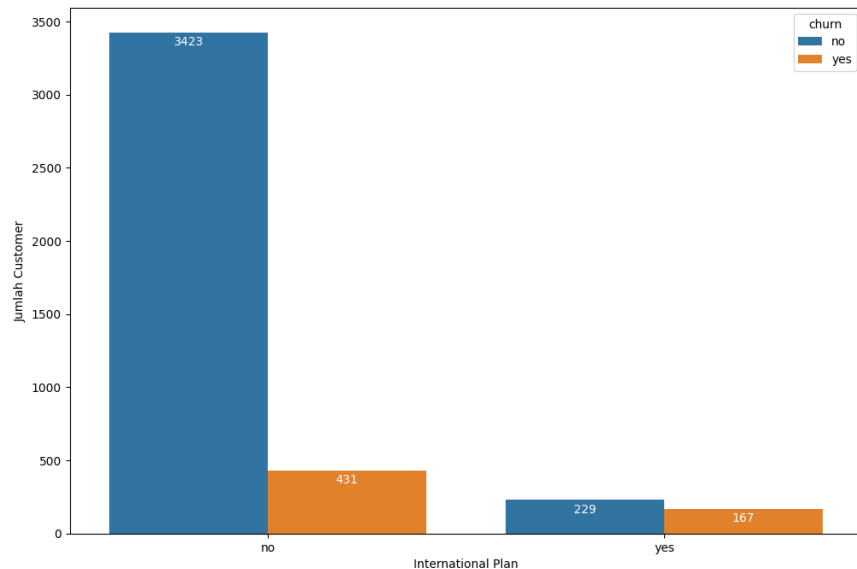
Customer yang berlangganan paket internasional hanya sebanyak 396 (9.31%) customer.



# Apakah berlangganan paket internasional berpengaruh terhadap customer churn?

## Berlangganan Paket Internasional Berpengaruh Terhadap Customer Churn

Hal ini menunjukkan bahwa berlangganan paket internasional meningkatkan persentase Churning, yaitu 42.17% dibandingkan dengan 11.18%



## Insight

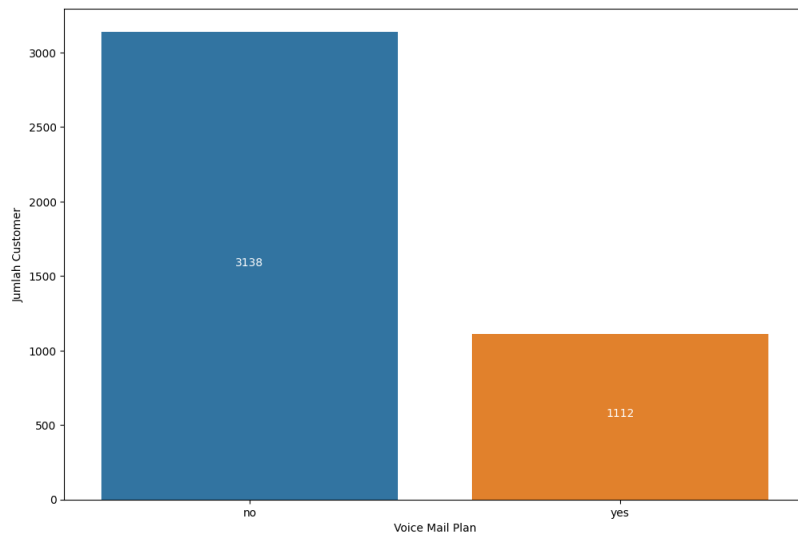
Walaupun jumlah customer churn pada customer yang tidak berlangganan paket internasional mencapai 431 customer, namun ditemukan bahwa persentase customer churn lebih tinggi pada customer yang berlangganan paket internasional. Dengan demikian dapat disimpulkan bahwa berlangganan paket internasional dapat mempengaruhi customer churn.



# Berapa jumlah customer yang berlangganan paket voice mail?

## Jumlah Customer Berdasarkan Paket Voice Mail

Customer yang berlangganan paket Voice Mail hanya berjumlah 1.112 (26.16%) Customer



## Insight

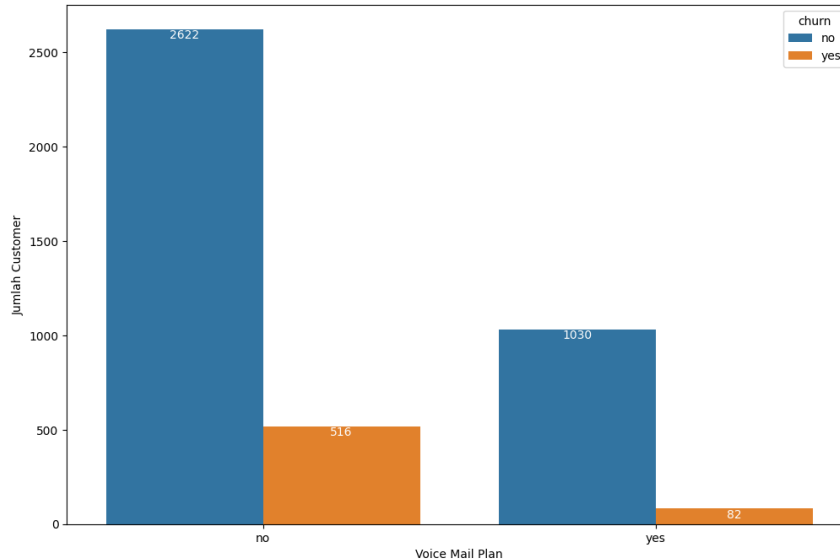
Customer yang berlangganan paket voice mail hanya sebanyak 1.112 (26.16%) customer.



# Apakah berlangganan paket voice mail berpengaruh terhadap customer churn?

## Berlangganan Paket Voice Mail Tidak Banyak Berpengaruh Terhadap Customer Churn

Hal ini menunjukkan bahwa berlangganan paket Voice Mail mengurangi persentase Churning, yaitu 7.37% dibandingkan dengan 16.44%



## Insight

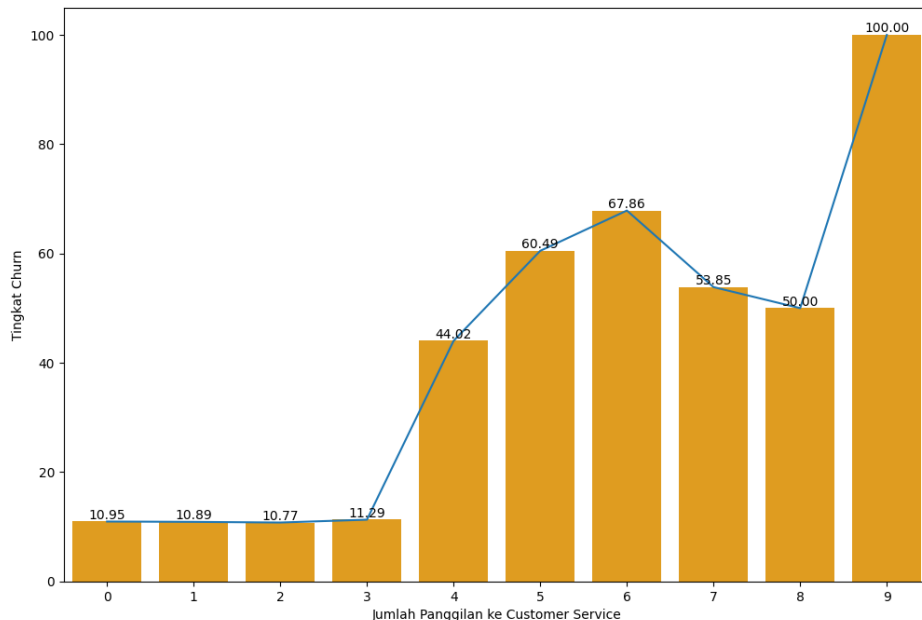
Walaupun retensi customer banyak ditemukan pada customer yang tidak berlangganan paket voice mail, namun customer yang tidak berlangganan paket voice mail lebih banyak melakukan churn daripada customer yang berlangganan paket voice mail. Oleh karena itu dapat disimpulkan bahwa berlangganan paket voice mail cenderung tidak mempengaruhi customer churn.



# Apakah jumlah panggilan ke customer service berpengaruh pada customer churn?

## Banyaknya panggilan ke customer service berpengaruh terhadap customer churn

Hal ini menunjukkan bahwa banyaknya panggilan ke customer service cenderung meningkatkan persentase customer churn

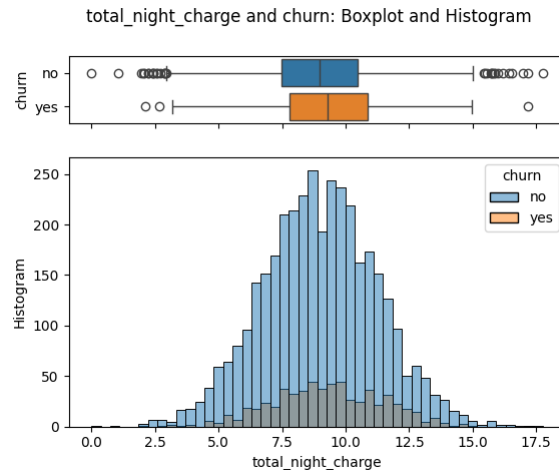
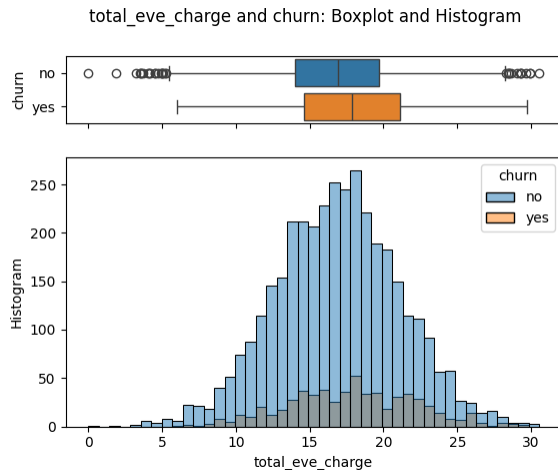
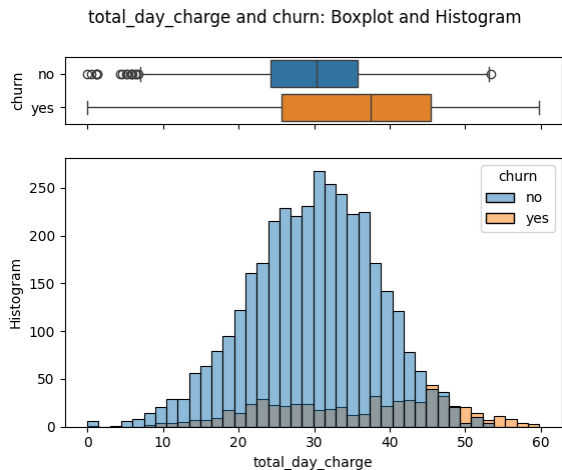


## Insight

Banyaknya jumlah panggilan ke customer service berpengaruh terhadap customer churn, hal ini terlihat ketika panggilan berjumlah di atas 3 kali terjadi peningkatan tingkat churn yang signifikan dan seiring bertambahnya jumlah panggilan seterusnya, tingkat churn relatif meningkat.



# Apakah beban/biaya yang dikeluarkan customer berpengaruh pada customer churn?



## Insight

Semakin besar biaya panggilan yang dibebankan ke customer maka semakin berpotensi customer dapat melakukan churn.



A group of people are seated in a room, facing towards the right. In the background, a large, stylized logo for 'BIMAR' is visible on the wall. The scene is overlaid with a semi-transparent purple filter. The word 'Preprocessing' is centered in the image in a white, bold, sans-serif font.

# Preprocessing

## Cek Missing Value & Duplicate Data

```
print(f'Jumlah missing value: \n{df_train.isnull().sum()}')
```

```
Jumlah missing value:  
state                0  
account_length      0  
area_code           0  
international_plan  0  
voice_mail_plan     0  
number_vmail_messages 0  
total_day_minutes   0  
total_day_calls     0  
total_day_charge     0  
total_eve_minutes   0  
total_eve_calls     0  
total_eve_charge    0  
total_night_minutes 0  
total_night_calls   0  
total_night_charge  0  
total_intl_minutes  0  
total_intl_calls    0  
total_intl_charge   0  
number_customer_service_calls 0  
churn               0  
dtype: int64
```

```
[ ] #mengecek apakah ada data yg terduplikat  
print(f'Jumlah data terduplikat:\n{df_train[df_train.duplicated()].sum()}')
```

```
Jumlah data terduplikat:  
state                0  
account_length      0  
area_code           0  
international_plan  0  
voice_mail_plan     0  
number_vmail_messages 0  
total_day_minutes   0.0  
total_day_calls     0  
total_day_charge     0.0  
total_eve_minutes   0.0  
total_eve_calls     0  
total_eve_charge    0.0  
total_night_minutes 0.0  
total_night_calls   0  
total_night_charge  0.0  
total_intl_minutes  0.0  
total_intl_calls    0  
total_intl_charge   0.0  
number_customer_service_calls 0  
churn               0  
dtype: object
```





## Cek Outliers

```
[ ] #detect outlier dengan IQR
```

```
def detect_outlier(data, kolom):  
    Q1 = data[kolom].quantile(0.25)  
    Q3 = data[kolom].quantile(0.75)
```

```
    IQR = Q3-Q1
```

```
    batas_bawah = Q1 - 1.5 * IQR  
    batas_atas = Q3 + 1.5 * IQR
```

```
    return data[(data[kolom] < batas_bawah)|(data[kolom] > batas_atas)].shape[0]
```

```
[ ] outlier = []
```

```
for kolom in kolom_numerik.columns:
```

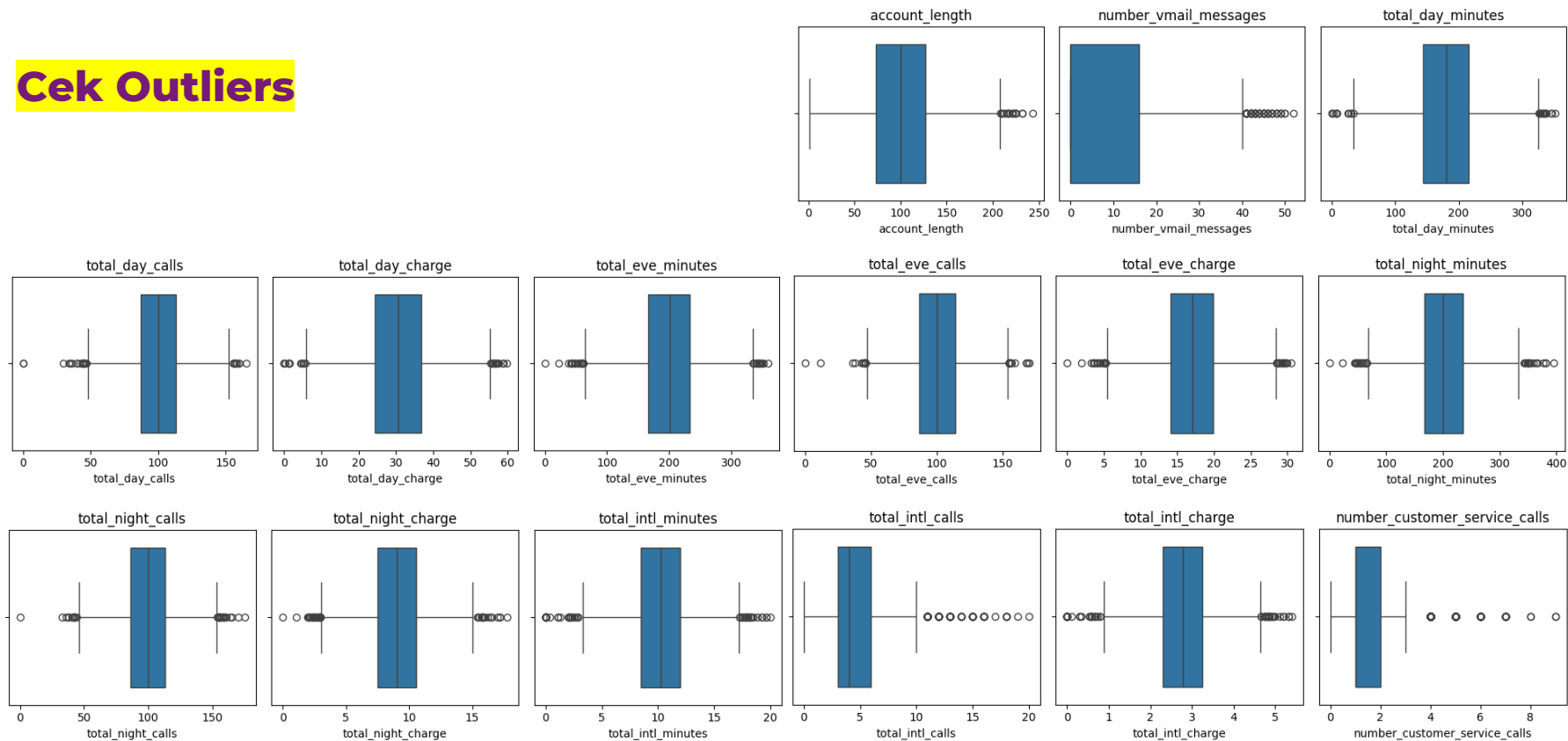
```
    outlier.append(f'{kolom}, jumlah outliers: {detect_outlier(df_train, kolom)}, persentase: {((detect_outlier(df_train, kolom)/len(df_train[kolom]))*100)}')
```

```
outlier
```

```
[ 'account_length, jumlah outliers: 20, persentase: 0.4705882352941176',  
  'number_vmail_messages, jumlah outliers: 86, persentase: 2.023529411764706',  
  'total_day_minutes, jumlah outliers: 25, persentase: 0.5882352941176471',  
  'total_day_calls, jumlah outliers: 28, persentase: 0.6588235294117647',  
  'total_day_charge, jumlah outliers: 26, persentase: 0.611764705882353',  
  'total_eve_minutes, jumlah outliers: 34, persentase: 0.8',  
  'total_eve_calls, jumlah outliers: 24, persentase: 0.5647058823529412',  
  'total_eve_charge, jumlah outliers: 34, persentase: 0.8',  
  'total_night_minutes, jumlah outliers: 37, persentase: 0.8705882352941177',  
  'total_night_calls, jumlah outliers: 33, persentase: 0.7764705882352941',  
  'total_night_charge, jumlah outliers: 37, persentase: 0.8705882352941177',  
  'total_intl_minutes, jumlah outliers: 62, persentase: 1.4588235294117646',  
  'total_intl_calls, jumlah outliers: 100, persentase: 2.3529411764705883',  
  'total_intl_charge, jumlah outliers: 62, persentase: 1.4588235294117646',  
  'number_customer_service_calls, jumlah outliers: 335, persentase: 7.882352941176471']
```



# Cek Outliers



# Handling Outliers

```
[ ] df_ubah = df_train.copy()
```

```
[ ] def ubah_outlier(kolom):  
    Q1 = df_ubah[kolom].quantile(0.25)  
    Q3 = df_ubah[kolom].quantile(0.75)  
    IQR = Q3-Q1  
    batas_bawah = Q1-1.5*IQR  
    batas_atas = Q3+1.5*IQR  
    df_ubah[kolom] = np.where(df_ubah[kolom] < batas_bawah, batas_bawah, df_ubah[kolom])  
    df_ubah[kolom] = np.where(df_ubah[kolom] > batas_atas, batas_atas, df_ubah[kolom])
```

```
[ ] for kolom in kolom_numerik.columns:  
    ubah_outlier(kolom)
```

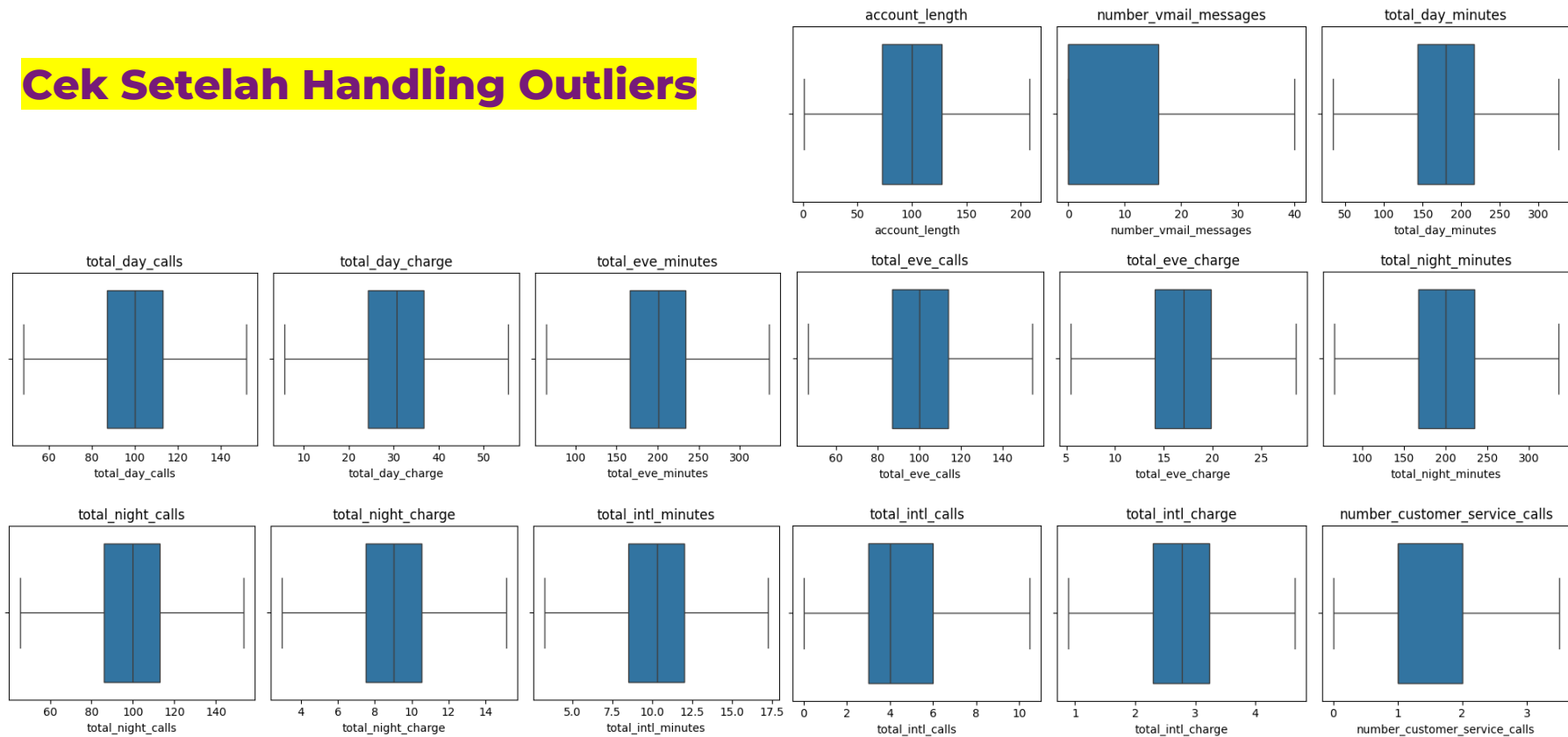
```
[ ] re_check_outlier = []  
  
for kolom in kolom_numerik.columns:  
    re_check_outlier.append(f'{kolom}, jumlah outliers: {detect_outlier(df_ubah, kolom)}')  
  
re_check_outlier
```

## Output:

```
⇒ ['account_length, jumlah outliers: 0',  
   'number_vmail_messages, jumlah outliers: 0',  
   'total_day_minutes, jumlah outliers: 0',  
   'total_day_calls, jumlah outliers: 0',  
   'total_day_charge, jumlah outliers: 0',  
   'total_eve_minutes, jumlah outliers: 0',  
   'total_eve_calls, jumlah outliers: 0',  
   'total_eve_charge, jumlah outliers: 0',  
   'total_night_minutes, jumlah outliers: 0',  
   'total_night_calls, jumlah outliers: 0',  
   'total_night_charge, jumlah outliers: 0',  
   'total_intl_minutes, jumlah outliers: 0',  
   'total_intl_calls, jumlah outliers: 0',  
   'total_intl_charge, jumlah outliers: 0',  
   'number_customer_service_calls, jumlah outliers: 0']
```



## Cek Setelah Handling Outliers



# Standarisasi/Normalisasi

Pada tahap preprocessing akan menggunakan data ber-outliers (df\_train) dan data tanpa outliers (df\_cleaned). Tujuannya adalah untuk membandingkan model ketika di uji untuk melihat seberapa berpengaruhnya nilai outliers terhadap performa model.

```
[ ] from sklearn.preprocessing import StandardScaler
```


```
[ ] def standarisasi(data):  
    scaler = StandardScaler()  
    data_standarisasi = pd.DataFrame(scaler.fit_transform(data.loc[:, kolom_numerik.columns].values), columns=kolom_numerik.columns)  
    return pd.concat([data_standarisasi, data[['state', 'area_code', 'international_plan', 'voice_mail_plan', 'churn']]], axis=1)
```

**df\_train**

```
[ ] df_train = standarisasi(df_train)
```


```
[ ] df_train.head()
```

**Sebelum**



	account_length	number_vmail_messages	total_day_minutes
0	107	26	161.6
1	137	0	243.4

**df\_train**



	account_length	number_vmail_messages	total_day_minutes
0	0.170399	1.366857	-0.345510
1	0.926186	-0.567911	1.169136


**df\_cleaned**

```
[ ] df_cleaned = standarisasi(df_cleaned)
```

```
[ ] df_cleaned.head()
```

**Sesudah**

**df\_cleaned**



	account_length	number_vmail_messages	total_day_minutes
0	0.172555	1.393905	-0.347524
1	0.931543	-0.570844	1.174129



# Feature Encoding

Pada tahap ini kolom kategorik akan diubah nilainya menjadi numerik menggunakan **Label Encoder** dan **One-Hot Encoder**.

```
[ ] from sklearn import preprocessing
    from sklearn.preprocessing import LabelEncoder, OneHotEncoder

[ ] def label_encode(data):
    le = LabelEncoder()

    for i in kolom_kategorik.columns:
        data[i] = le.fit_transform(data[i])
    return data

[ ] def onehot(data):
    return pd.get_dummies(data, columns=['area_code'], dtype=int, prefix='area_code')
```

## df\_train

```
[ ] df_train = label_encode(df_train)
    df_train = onehot(df_train)
    df_train = df_train.rename(columns={'area_code_0': 'area_code_408', 'area_code_1': 'area_code_415', 'area_code_2': 'area_code_510'})
```

## df\_cleaned

```
[ ] df_cleaned = label_encode(df_cleaned)
    df_cleaned = onehot(df_cleaned)
    df_cleaned = df_cleaned.rename(columns={'area_code_0': 'area_code_408', 'area_code_1': 'area_code_415', 'area_code_2': 'area_code_510'})
```



## Hasil Feature Encoding

### Sebelum

area_code	international_plan	voice_mail_plan
area_code_415	no	yes
area_code_415	no	no
area_code_408	yes	no
area_code_415	yes	no
area_code_510	no	yes

### Sesudah

international_plan	voice_mail_plan	area_code_408	area_code_415	area_code_510
0	1	0	1	0
0	0	0	1	0
1	0	1	0	0
1	0	0	1	0
0	1	0	0	1



## Balancing Data

```
[ ] from imblearn.over_sampling import SMOTE
```

```
[ ] df_train['churn'].value_counts()
```

```
↔ churn
0    3652
1     598
Name: churn, dtype: int64
```

Sesudah

```
↔ churn
0    3652
1    3652
Name: churn, dtype: int64
```

```
[ ] x_df_train = df_train.drop(columns='churn')
   y_df_train = df_train.loc[:, 'churn']
```

```
[ ] sampler = SMOTE(random_state=0)
   x_res_df_train, y_res_df_train = sampler.fit_resample(x_df_train, y_df_train)
```

Pada df\_cleaned juga akan dilakukan balancing data dengan cara yang sama.





## Split Data (80:20)

### df\_train

```
[ ] from sklearn.model_selection import train_test_split
```

```
[ ] x_train_df_train, x_test_df_train, y_train_df_train, y_test_df_train = train_test_split(x_res_df_train, y_res_df_train, test_size=0.2, random_state=42)
```

```
➦ jumlah baris x_train: 5843, jumlah kolom x_train: 20  
jumlah baris x_test: 1461, jumlah kolom x_test: 20  
jumlah baris y_train: 5843, jumlah kolom y_train: 1  
jumlah baris y_test: 1461, jumlah kolom y_test: 1
```

### df\_cleaned

```
[ ] x_train_df_cleaned, x_test_df_cleaned, y_train_df_cleaned, y_test_df_cleaned = train_test_split(x_res_df_cleaned, y_res_df_cleaned, test_size=0.2, random_state=42)
```

```
➦ jumlah baris x_train: 5843, jumlah kolom x_train: 20  
jumlah baris x_test: 1461, jumlah kolom x_test: 20  
jumlah baris y_train: 5843, jumlah kolom y_train: 1  
jumlah baris y_test: 1461, jumlah kolom y_test: 1
```



A group of people are sitting in a circle in a room. In the background, there is a large sign that says 'BIMAR' in a stylized, outlined font. The room has a checkered floor and some decorations. The word 'Modeling' is overlaid in the center of the image.

# Modeling

# Random Forest

```
[ ] from sklearn.ensemble import RandomForestClassifier  
    from sklearn.metrics import accuracy_score
```

## df\_train

```
[ ] model_randomforest_df_train = RandomForestClassifier()
```

```
[ ] model_randomforest_df_train.fit(x_train_df_train, y_train_df_train)
```

## df\_cleaned

```
[ ] model_randomforest_df_cleaned = RandomForestClassifier()
```

```
[ ] model_randomforest_df_cleaned.fit(x_train_df_cleaned, y_train_df_cleaned)
```



# Decision Tree

```
[ ] from sklearn.tree import DecisionTreeClassifier
```

## df\_train

```
[ ] model_decisiontree_df_train = DecisionTreeClassifier(random_state=42)
```

```
[ ] model_decisiontree_df_train.fit(x_train_df_train, y_train_df_train)
```

## df\_cleaned

```
[ ] model_decisiontree_df_cleaned = DecisionTreeClassifier(random_state=42)
```

```
[ ] model_decisiontree_df_cleaned.fit(x_train_df_cleaned, y_train_df_cleaned)
```



# Support Vector Machine

```
[ ] from sklearn.svm import SVC
```

## df\_train

```
[ ] model_svm_df_train = SVC(random_state=42)
```

```
[ ] model_svm_df_train.fit(x_train_df_train, y_train_df_train)
```

## df\_cleaned

```
[ ] model_svm_df_cleaned = SVC(random_state=42)
```

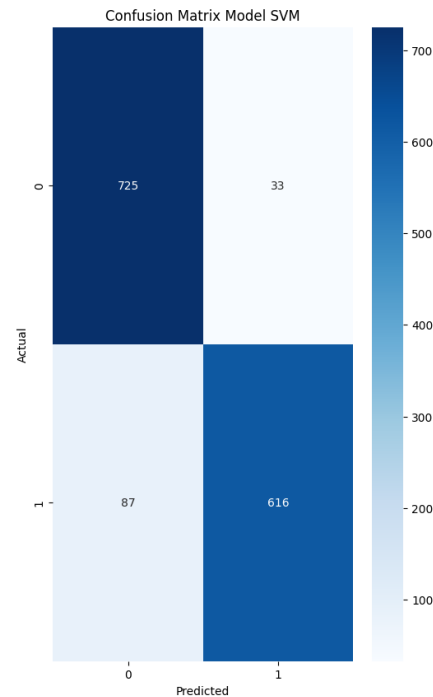
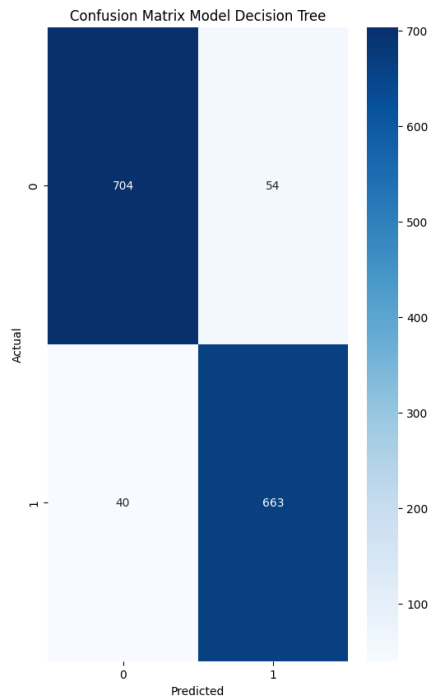
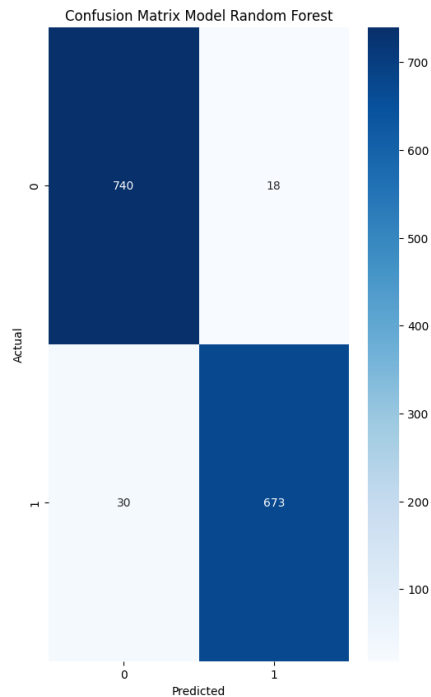
```
[ ] model_svm_df_cleaned.fit(x_train_df_cleaned, y_train_df_cleaned)
```



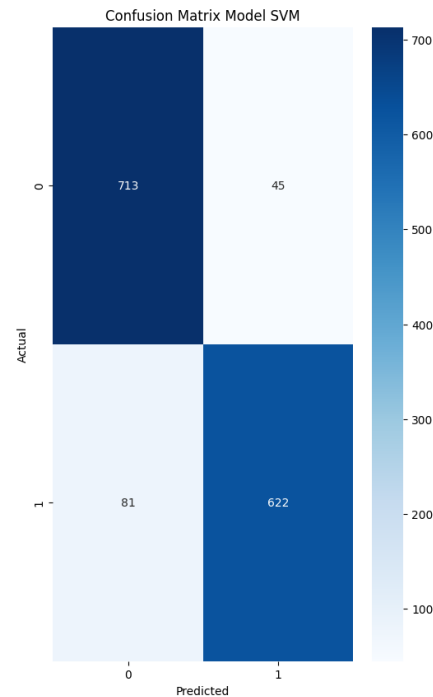
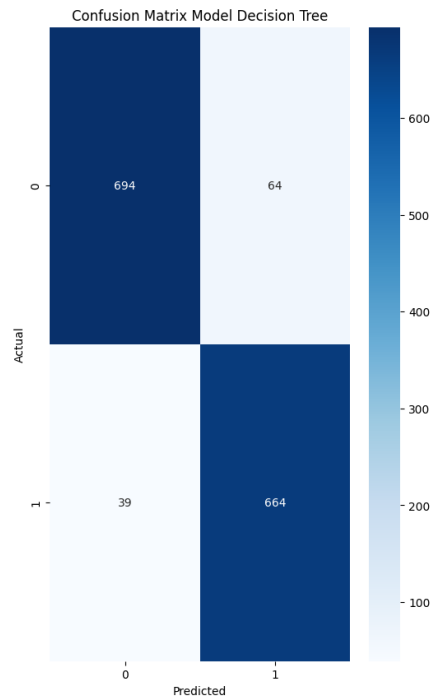
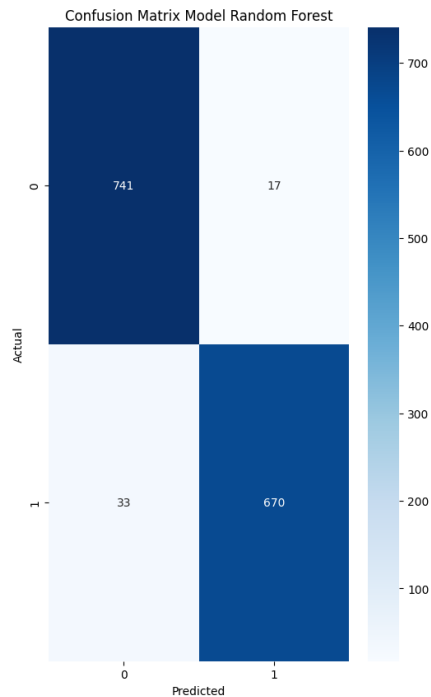
A group of people are sitting in a circle in a room. In the background, there is a large wall with the word 'BIMAR' written in large, stylized letters. The room has a modern, industrial feel with large windows and a concrete floor. The people are dressed in casual to semi-formal attire. The word 'Evaluation' is overlaid in the center of the image.

# Evaluation

## Confusion Matrix (df\_train)



## Confusion Matrix (df\_cleaned)





## Classification Report (df\_train)

Classification	Report Model	Random Forest:		
	precision	recall	f1-score	support
0	0.9610	0.9763	0.9686	758
1	0.9740	0.9573	0.9656	703
accuracy			0.9671	1461
macro avg	0.9675	0.9668	0.9671	1461
weighted avg	0.9673	0.9671	0.9671	1461

Classification	Report Model	SVM:		
	precision	recall	f1-score	support
0	0.8929	0.9565	0.9236	758
1	0.9492	0.8762	0.9112	703
accuracy			0.9179	1461
macro avg	0.9210	0.9164	0.9174	1461
weighted avg	0.9199	0.9179	0.9176	1461

Classification	Report Model	Decision Tree:		
	precision	recall	f1-score	support
0	0.9462	0.9288	0.9374	758
1	0.9247	0.9431	0.9338	703
accuracy			0.9357	1461
macro avg	0.9355	0.9359	0.9356	1461
weighted avg	0.9359	0.9357	0.9357	1461



## Classification Report (df\_cleaned)

Classification Report Model Random Forest:					
	precision	recall	f1-score	support	
0	0.9574	0.9776	0.9674	758	
1	0.9753	0.9531	0.9640	703	
accuracy			0.9658	1461	
macro avg	0.9663	0.9653	0.9657	1461	
weighted avg	0.9660	0.9658	0.9658	1461	

Classification Report Model SVM:					
	precision	recall	f1-score	support	
0	0.8980	0.9406	0.9188	758	
1	0.9325	0.8848	0.9080	703	
accuracy			0.9138	1461	
macro avg	0.9153	0.9127	0.9134	1461	
weighted avg	0.9146	0.9138	0.9136	1461	

Classification Report Model Decision Tree:					
	precision	recall	f1-score	support	
0	0.9468	0.9156	0.9309	758	
1	0.9121	0.9445	0.9280	703	
accuracy			0.9295	1461	
macro avg	0.9294	0.9300	0.9295	1461	
weighted avg	0.9301	0.9295	0.9295	1461	



A group of people are sitting in a circle in a room. In the background, there is a large sign that says 'BIMAR' in a stylized, outlined font. The room has a checkered floor and some decorations. The word 'Prediction' is overlaid in the center of the image.

# Prediction

## Random Forest

id prediksi_churn		
0	1	no
1	2	no
2	3	yes
3	4	no
4	5	no
...	...	...
745	746	no
746	747	no
747	748	no
748	749	no
749	750	no

750 rows × 2 columns

## Decision Tree

id prediksi_churn		
0	1	no
1	2	no
2	3	yes
3	4	yes
4	5	yes
...	...	...
745	746	no
746	747	yes
747	748	yes
748	749	yes
749	750	no

750 rows × 2 columns

## SVM

id prediksi_churn		
0	1	no
1	2	no
2	3	no
3	4	no
4	5	no
...	...	...
745	746	no
746	747	no
747	748	no
748	749	no
749	750	no

750 rows × 2 columns



## df\_cleaned

### Random Forest

id prediksi_churn		
0	1	no
1	2	no
2	3	yes
3	4	no
4	5	no
...	...	...
745	746	no
746	747	no
747	748	no
748	749	no
749	750	no

750 rows x 2 columns

### Decision Tree

id prediksi_churn		
0	1	no
1	2	no
2	3	yes
3	4	yes
4	5	yes
...	...	...
745	746	no
746	747	yes
747	748	yes
748	749	no
749	750	no

750 rows x 2 columns

### SVM

id prediksi_churn		
0	1	no
1	2	no
2	3	yes
3	4	no
4	5	no
...	...	...
745	746	no
746	747	no
747	748	no
748	749	no
749	750	no

750 rows x 2 columns





## Kesimpulan

## Kesimpulan

- Berdasarkan hasil performa masing-masing model telah menunjukkan bahwa data ber-outliers tingkat akurasi lebih tinggi daripada data tanpa outliers. Hal ini dapat disimpulkan bahwa adanya outliers belum tentu dapat memberikan hasil yang tidak akurat, bisa jadi adanya outliers dapat memperkaya data sehingga performa yang dihasilkan model dapat lebih baik.
- Model Random Forest menghasilkan tingkat akurasi sebesar 96.71% pada data ber-outliers dan 96.58% pada data tanpa outliers menjadikannya model dengan performa paling baik dari pada model Decision Tree dan SVM.



## Ringkasan Insight

1. Tingkat churn tertinggi terdapat di negara bagian New Jersey, sementara itu negara bagian West Virginia menjadi negara bagian dengan jumlah retensi customer tertinggi.
2. Berlangganan paket internasional berpengaruh terhadap customer churn.
3. Berlangganan paket voice mail cenderung tidak berpengaruh terhadap customer churn.
4. Banyaknya jumlah panggilan ke customer service dapat mempengaruhi customer churn.
5. Semakin besar biaya panggilan yang dibebankan ke customer maka semakin berpotensi customer dapat melakukan churn.





## Rekomendasi

1. Perusahaan perlu mempertimbangkan untuk memberikan diskon dengan periode tertentu pada customer di beberapa negara bagian dengan tingkat churn yang tinggi dan perlu juga untuk memberikan bonus dengan periode tertentu pada customer di negara bagian dengan jumlah retensi customer yang tinggi. Sehingga harapannya dengan memberikan diskon dan bonus dapat memperkecil tingkat churn dan menjaga retensi customer.
2. Perusahaan perlu memastikan layanan paket internasional yang diberikan memenuhi atau melebihi harapan customer. Hal ini mencakup aspek-aspek seperti keandalan jaringan, kecepatan koneksi, dan layanan customer yang responsif. Hal ini dikarenakan banyak customer yang berlangganan paket internasional cenderung untuk melakukan churn.
3. Perusahaan perlu memberikan diskon atau bonus kepada customer yang berlangganan paket voice mail selama periode tertentu. Hal ini dikarenakan customer yang berlangganan paket voice mail cenderung untuk bertahan, sehingga dengan memberikan diskon atau bonus harapannya dapat menjaga retensi customer.



## Rekomendasi

4. Perusahaan perlu memperbaiki kualitas layanan customer service kepada customer agar lebih responsif dan dapat memberikan solusi yang tepat, sehingga customer akan merasa lebih terbantu dan masalah yang dihadapi customer dapat terselesaikan dengan baik. Sehingga dengan adanya perbaikan terhadap layanan customer service, harapannya customer menjadi lebih puas terhadap layanan yang diberikan dan dapat menjaga retensi customer.
5. Perusahaan perlu mempertimbangkan biaya panggilan yang dibebankan ke customer agar biaya panggilan lebih terjangkau. Hal ini dikarenakan semakin besar biaya panggilan yang dibebankan ke customer maka semakin berpotensi customer melakukan churn.



Terima kasih!



Contact:

