

# Prediksi *Customer Churn* Menggunakan Algoritma Decision Tree

Diterima:

10 Juni 2024

Revisi:

10 Juli 2024

Terbit:

1 Agustus 2024

<sup>1\*</sup>Dewi Putriani, <sup>2</sup>Anindita Puspa Ayu Prayogi, <sup>3</sup>Altha Inas  
Shofyana, <sup>4</sup>Aidina Ristyawan, <sup>5</sup>Erna Daniati

<sup>1-5</sup>Universitas Nusantara PGRI Kediri

<sup>1</sup>[dewiptr098@gmail.com](mailto:dewiptr098@gmail.com) · <sup>2</sup>[aninditapuspa3@gmail.com](mailto:aninditapuspa3@gmail.com),

<sup>3</sup>[althainas02@gmail.com](mailto:althainas02@gmail.com) · <sup>4</sup>[aidinaristi@unpkediri.ac.id](mailto:aidinaristi@unpkediri.ac.id),

<sup>5</sup>[ernadaniati@unpkediri.ac.id](mailto:ernadaniati@unpkediri.ac.id)

**Abstrak**—Dalam era bisnis modern, mempertahankan pelanggan yang ada seringkali lebih menguntungkan dibanding menarik pelanggan baru. Salah satu tantangan terbesar adalah churn atau hilangnya pelanggan. Prediksi churn menggunakan data mining, khususnya algoritma decision tree, memungkinkan perusahaan untuk mengidentifikasi faktor-faktor penyebab churn dan mengklasifikasikan pelanggan berdasarkan kemungkinan mereka untuk berhenti berlangganan. Penelitian ini menggunakan metodologi SEMMA (*Sample, Explore, Modify, Model, Assess*) untuk menganalisis dataset customer churn yang terdiri dari 64.000 data. Hasil penelitian menunjukkan bahwa model decision tree memiliki performa prediksi lebih tinggi dibandingkan algoritma lainnya, dengan akurasi mencapai 99.7%, *precision* 99.7%, *recall* 99.7%, dan *F1-score* 99.7%. Temuan ini memberikan wawasan penting bagi perusahaan dalam mengembangkan strategi retensi pelanggan yang lebih efektif.

**Kata Kunci**—customer churn; decision tree; data mining; SEMMA

**Abstract**—In the modern business era, retaining existing customers is often more profitable than attracting new ones. One of the biggest challenges is churn, or customer loss. Predicting churn using data mining, particularly the decision tree algorithm, enables companies to identify the factors causing churn and classify customers based on their likelihood of leaving. This study employs the SEMMA (*Sample, Explore, Modify, Model, Assess*) methodology to analyze a customer churn dataset comprising 64,000 records. The results indicate that the decision tree model achieves higher predictive performance compared to other algorithms, with an accuracy of 99.7%, precision of 99.7%, recall of 99.7%, and F1-score of 99.7%. These findings provide valuable insights for companies to develop more effective customer retention strategies.

**Keywords**—customer churn; decision tree; data mining; SEMMA

This is an open access article under the CC BY-SA License.



---

## Penulis Korespondensi:

Aidina Ristyawan,  
Sistem Informasi Fakultas Teknik dan Ilmu Komputer,  
Universitas Nusantara PGRI Kediri,  
Email: [aidinaristi@unpkediri.ac.id](mailto:aidinaristi@unpkediri.ac.id)  
ID Orcid: [ <https://orcid.org/0009-0003-2712-1507> ]  
Handphone: 081232624460

---

## I. PENDAHULUAN

Dalam era bisnis modern, mempertahankan pelanggan yang ada seringkali lebih menguntungkan dibanding menarik pelanggan baru dan juga membantu perusahaan menjual lebih banyak produk [1], [2]. Masalah yang dihadapi oleh perusahaan adalah bagaimana mencegah fenomena customer churn ini, yang terjadi ketika pelanggan berhenti berlangganan layanan atau produk perusahaan [3], [4]. Fenomena ini menjadi masalah kritis karena dapat menyebabkan penurunan pendapatan dan mengganggu stabilitas bisnis perusahaan. Untuk mengatasi masalah ini, diperlukan metode prediksi yang efektif. Prediksi customer churn bertujuan untuk mengklasifikasikan data pelanggan sebelumnya menjadi dua kategori: pelanggan yang akan berhenti berlangganan dan pelanggan yang akan terus berlangganan [5], [6]. Prediksi tersebut memanfaatkan ilmu data mining peran klasifikasi yang merupakan menempatkan variabel atau objek ke dalam beberapa kategori relevan yang telah ditetapkan sebelumnya [4], [7]. Dalam proses eksekusi data mining, diperlukan sebuah algoritma yang dapat mengklasifikasikan apakah *customer churn* atau tidak *churn* [8]. Untuk itu dalam proses prediksi *churn* algoritma yang digunakan adalah *decision tree*. Algoritma *decision tree* adalah salah satu algoritma yang sangat populer pada peran data mining klasifikasi strukturnya berbentuk pohon dimana setiap *node* merepresentasikan atribut yang diuji, dan setiap cabang dari *node decision tree* merepresentasikan hasil dari uji atribut tersebut [6]. Analisis ini memungkinkan perusahaan untuk mengadaptasi strategi pemasaran, layanan pelanggan, dan pengembangan produk sesuai dengan kebutuhan pelanggan, sehingga meningkatkan retensi, mengurangi *churn*, dan loyalitas pelanggan [9].

Penelitian ini menggunakan metodologi SEMMA (*Sample, Explore, Modify, Model, Access*) [6]. SEMMA adalah kerangka kerja yang sistematis dan terstruktur yang membantu dalam setiap tahap analisis data, mulai dari pengambilan sampel hingga evaluasi model [10]. SEMMA memberikan panduan analisis data dan pembangunan model prediktif dengan menekankan eksplorasi data, modifikasi data, penggunaan model yang tepat, dan evaluasi menyeluruh untuk memastikan performa model yang optimal dan handal [11]. Penelitian ini bertujuan untuk mengevaluasi performa dari peneliti terdahulu A. Yulianto and Firmansyah (2021), yang menggunakan algoritma *naive bayes* dan mencapai hasil akurasi sebesar 80% [12]. Meskipun telah dilakukan beberapa penelitian untuk memprediksi churn pelanggan, masih ada kebutuhan untuk menemukan metode yang lebih akurat dan efisien. Penelitian sebelumnya menunjukkan bahwa algoritma naive bayes hanya mencapai akurasi sebesar 80%, yang menunjukkan bahwa masih ada ruang untuk peningkatan akurasi prediksi. Oleh karena itu, diperlukan penelitian lebih lanjut untuk mengevaluasi dan meningkatkan akurasi model prediksi churn dengan menggunakan algoritma yang lebih kuat. Selanjutnya untuk mengetahui algoritma

yang memiliki akurasi tertinggi dilakukan perbandingan antar algoritma yaitu naive bayes, k-nearest neighbor, dan decision tree.

Tabel 1. Hasil Perbandingan Algoritma

Algoritma	Akurasi	Presisi
Naive Bayes	86,83%	87,41%
KNN	78.69%	85.06%
Decision Tree	99.78%	99.77%

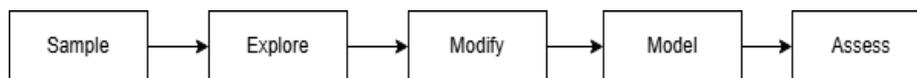
Tabel 1 di atas menunjukkan hasil perbandingan kinerja tiga algoritma yang berbeda dalam memprediksi churn pelanggan, yaitu naive bayes, k-nearest neighbor (KNN), dan decision tree.

Dua metrik utama yang digunakan untuk mengevaluasi kinerja algoritma tersebut adalah akurasi dan presisi. Dari tabel tersebut, dapat dilihat bahwa algoritma decision tree memiliki kinerja yang paling tinggi dibandingkan dengan naive bayes dan KNN. Decision tree mencapai akurasi sebesar 99.78% dan presisi sebesar 99.77%, jauh melampaui naive bayes yang memiliki akurasi 86.83% dan presisi 87.41%, serta KNN yang memiliki akurasi 78.69% dan presisi 85.06%.

Penelitian ini diharapkan dapat memberikan beberapa manfaat, antara lain: peningkatan akurasi prediksi churn, pengembangan strategi retensi pelanggan yang lebih efektif, pengembangan model prediktif yang handal, dan kontribusi terhadap literasi data mining.

## II. METODE

Metode yang digunakan dalam penelitian ini adalah proses data mining SEMMA. Proses SEMMA (*Sample, Explore, Modify, Model, Assess*) memiliki keunggulan yaitu proses yang memberikan penekanan yang lebih besar pada pengembangan model dalam data mining, sehingga proses pembuatan model menjadi lebih terorganisir[14].



Gambar 1. Metodologi Penelitian

Penjelasan mengenai alur metodologi penelitian adalah sebagai berikut.

1. *Sample* dalam tahap ini menentukan sampel data yang akan dianalisis, menggunakan dataset *public customer churn* sebanyak 64.000 data dari Kaggle untuk eksplorasi dan *preprocessing* [13], [14].
2. *Explore* (Deskripsi data) dalam tahap ini adalah mendeskripsikan data - data yang digunakan untuk mengetahui kondisi dan gambaran besar informasi yang ada dalam data tersebut [13].

3. *Modify* tahap ini melibatkan penyesuaian data melalui pemilihan dan perubahan tipe variabel, *data cleaning*, dan preprocessing untuk memfokuskan proses pemodelan dan memudahkan pengolahan data [15].
4. *Model* tahap ini melibatkan penerapan algoritma *decision tree* untuk klasifikasi dan visualisasi faktor-faktor yang mempengaruhi perusahaan kehilangan pelanggan [15].
5. *Assess* tahap ini melibatkan evaluasi hasil pemodelan data dengan menampilkan nilai *accuracy*, *precision*, *recall*, dan *F1 Score*, serta mengukur *error* melalui nilai loss yang semakin rendah menandakan model yang lebih baik [13].

### III. HASIL DAN PEMBAHASAN

Untuk memprediksi *customer churn* menggunakan bahasa pemrograman Python, diimplementasikan menggunakan Jupyter Notebook dengan beberapa *library* seperti Pandas, Numpy, Scikit-learn, Seaborn, dan Matplotlib. Proses ini mengikuti metodologi SEMMA: *Sample*, *Explore*, *Modify*, *Model*, dan *Assess*.

#### 1. *Sample*

Tahapan awal yang dilakukan adalah mengambil sampel dari dataset yang digunakan. Penelitian ini menggunakan sampel dataset *public* berisi data customer churn sebanyak 64374 data yang berasal dari Kaggle. Pada langkah ini, yang dilakukan adalah memuat dataset dari file csv dan menampilkan 5 data pertama pada dataset tersebut untuk memastikan bahwa data telah berhasil dimuat dengan benar. Implementasi pada Gambar

**Sample**

```
In [1]: import pandas as pd
df = pd.read_csv('customer_churn_dataset-testing-master.csv')
df.head()
```

Out[1]:

	CustomerID	Age	Gender	Tenure	Usage Frequency	Support Calls	Payment Delay	Subscription Type	Contract Length	Total Spend	Last Interaction	Churn
0	1	22	Female	25	14	4	27	Basic	Monthly	598	9	1
1	2	41	Female	28	28	7	13	Standard	Monthly	584	20	0
2	3	47	Male	27	10	2	29	Premium	Annual	757	21	0
3	4	35	Male	9	12	5	17	Premium	Quarterly	232	18	0
4	5	53	Female	58	24	9	2	Standard	Annual	533	18	0

Gambar 2. Sample

#### 2. *Explore*

Tahap kedua yaitu *Explore* dilakukan untuk memahami struktur, pola, dan anomali dalam data sehingga lebih mudah untuk mengetahui kondisi dan gambaran besar informasi yang ada dalam data tersebut.

```
Explore
In [2]: df.shape
Out[2]: (64374, 12)
In [3]: df.dtypes
Out[3]: CustomerID      int64
Age                  int64
Gender              object
Tenure              int64
Usage Frequency     int64
Support Calls       int64
Payment Delay       int64
Subscription Type   object
Contract Length     int64
Total Spend         int64
Last Interaction     int64
Churn               int64
dtype: object
```

Gambar 3. Explore

Pada gambar 3, dilakukan explore data dengan menampilkan shape atau ukuran dari dataframe yang menampilkan jumlah baris dan jumlah kolom. Pada cell selanjutnya adalah digunakan untuk menampilkan tipe data dari setiap kolom dalam dataframe tersebut. Pada data ini ada 12 kolom dengan 9 kolom bertipe data integer, dan 3 kolom bertipe data object.

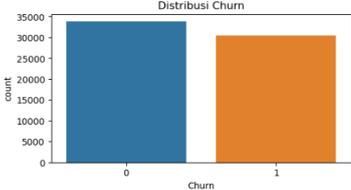
```
In [46]: df.describe()
Out[46]:
```

	CustomerID	Age	Tenure	Usage Frequency	Support Calls	Payment Delay	Total Spend	Last Interaction	Churn
count	64374.000000	64374.000000	64374.000000	64374.000000	64374.000000	64374.000000	64374.000000	64374.000000	64374.000000
mean	32187.500000	41.970982	31.994827	15.080234	5.400690	17.133952	541.023379	15.498850	0.473685
std	18583.317451	13.924911	17.098234	8.816470	3.114005	8.852211	260.874809	8.638436	0.499311
min	1.000000	18.000000	1.000000	1.000000	0.000000	0.000000	100.000000	1.000000	0.000000
25%	16094.250000	30.000000	18.000000	7.000000	3.000000	10.000000	313.000000	8.000000	0.000000
50%	32187.500000	42.000000	33.000000	15.000000	6.000000	19.000000	534.000000	15.000000	0.000000
75%	48280.750000	54.000000	47.000000	23.000000	8.000000	25.000000	768.000000	23.000000	1.000000
max	64374.000000	65.000000	60.000000	30.000000	10.000000	30.000000	1000.000000	30.000000	1.000000

Gambar 4. Describe

Selanjutnya pada gambar 4, dilakukan describe dataframe untuk menghasilkan ringkasan statistik deskriptif dari data numerik dalam DataFrame.

```
In [47]: import seaborn as sns
import matplotlib.pyplot as plt
categorical_vars = ['Churn']
for var in categorical_vars:
    plt.figure(figsize=(8, 3))
    sns.countplot(x=var)
    plt.title(f'Distribusi {var}')
    plt.show()
df['Churn'].value_counts()
```



```
Out[47]: Churn
0      33881
1      30493
Name: count, dtype: int64
```

Gambar 5. Distribusi Churn

Pada gambar 5 dilakukan visualisasi data mengenai penyebaran atau distribusi kelas dari churn dan didapatkan bahwa kelas dari dataset customer churn seimbang sehingga tidak memerlukan pemrosesan lebih lanjut.

```
In [50]: duplicates = df.duplicated()
print(f"Jumlah baris duplikat: {duplicates.sum()}")
Jumlah baris duplikat: 0

In [51]: print(df.isnull().sum())
CustomerID      0
Age              0
Gender           0
Tenure           0
Usage Frequency  0
Support Calls    0
Payment Delay    0
Subscription Type 0
Contract Length  0
Total Spend     0
Last Interaction 0
Churn            0
dtype: int64
```

Gambar 6. Data Duplikat

Pada gambar 6 dilakukan pengecekan data duplikat dan hasilnya menunjukkan bahwa dataset customer churn tidak memiliki data duplikat. Selanjutnya, dilakukan pengecekan data kosong dan hasil menunjukkan tiap kolom dalam dataset tidak memiliki nilai kosong (*null*).

### 3. Modify

#### Modify

```
In [52]: le = LabelEncoder()
df['Gender'] = le.fit_transform(df['Gender'])
df['Subscription Type'] = le.fit_transform(df['Subscription Type'])
df['Contract Length'] = le.fit_transform(df['Contract Length'])
print(df.head())
```

CustomerID	Age	Gender	Tenure	Usage Frequency	Support Calls	Payment Delay	Subscription Type	Contract Length	Total Spend	Last Interaction	Churn
0	1	22	0	25	14	4	0	1	598	9	1
1	2	41	0	28	7	1	1	1	584	20	0
2	3	47	1	27	2	0	1	0	757	21	0
3	4	35	1	9	12	2	1	2	232	18	0
4	5	53	0	58	24	0	2	0	533	18	0

Gambar 7. Modify

Pada gambar 7, hal pertama yang dilakukan adalah mengubah nilai atau isi dari 3 kolom (Gender, Subscription Type, dan Contract Length) yang bertipe object menjadi numerik agar bisa digunakan menjadi fitur untuk pemrosesan selanjutnya.

```
In [53]: X = df.drop(columns=['Churn'])
y = df['Churn']

In [54]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Gambar 8. Splitting Data

Selanjutnya, pada gambar 8 terdapat 2 cell. Cell pertama merupakan input untuk memisahkan fitur dan label sehingga menjadi X (berisi semua fitur kecuali kolom Churn) dan y (berisi label Churn). Cell kedua adalah proses pemisahan data latih dan data uji yang dilakukan dengan membagi menjadi 4 subset. Selanjutnya menentukan proporsi data yang akan digunakan sebagai data uji. Dalam kasus ini, 30% dari data akan digunakan sebagai data uji. Lalu

random\_state=42 yang digunakan untuk menetapkan keadaan awal dari generator angka acak, yang memastikan bahwa pembagian data menjadi data latih dan data uji akan konsisten.

#### 4. Model

```
Model  
  
In [55]: from sklearn.tree import DecisionTreeClassifier  
        dt_model = DecisionTreeClassifier()  
        dt_model.fit(X_train, y_train)  
  
Out[55]: DecisionTreeClassifier  
        DecisionTreeClassifier()
```

Gambar 9. Model

Berikutnya, pada gambar 9 merupakan pelatihan model *decision tree* menggunakan data latih yang sudah ditentukan sebelumnya yaitu sebesar 70% dari data keseluruhan. Model akan belajar dari hubungan antara fitur dan label untuk membuat prediksi. Agar mendapatkan hasil pelatihan yang konsisten, maka digunakan random\_state dengan parameter mengikuti proses *modify*.

#### 5. Assess

```
Asses  
  
In [74]: from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score  
        accuracy_dt = accuracy_score(y_test, y_pred_dt)  
        precision_dt = precision_score(y_test, y_pred_dt, average='macro')  
        recall_dt = recall_score(y_test, y_pred_dt, average='macro')  
        f1_dt = f1_score(y_test, y_pred_dt, average='macro')  
  
In [75]: print(f'Accuracy: {accuracy_dt}')  
        print(f'Precision: {precision_dt}')  
        print(f'Recall: {recall_dt}')  
        print(f'F1-score: {f1_dt}')  
  
Accuracy: 0.9978252990213845  
Precision: 0.9977950634893669  
Recall: 0.9978456350229938  
F1-Score: 0.9978200798118901  
  
In [76]: correct_predictions = (y_test == y_pred_dt).sum()  
        incorrect_predictions = len(y_test) - correct_predictions  
        total_test_samples = len(y_test)  
  
In [77]: print(f'Correct Predictions: {correct_predictions}')  
        print(f'Incorrect Predictions: {incorrect_predictions}')  
        print(f'Total Test Samples: {total_test_samples}')  
  
Correct Predictions: 19271  
Incorrect Predictions: 42  
Total Test Samples: 19313
```

Gambar 10. Assess

Pada gambar 10 menampilkan hasil evaluasi pemodelan data dengan metrik akurasi, presisi, recall, dan F1-score. Akurasi mengukur seberapa tepat model dalam mengklasifikasikan label, selanjutnya presisi mengukur proporsi prediksi positif yang benar dari semua prediksi semua prediksi positif, dan recall mengukur proporsi label positif yang benar dari semua label positif sebenarnya, serta F1-score rata - rata dari presisi dan recall yang memberikan gambaran umum tentang performa model.

Pada cell kedua adalah menampilkan hasil perhitungan dari masing-masing metrik yaitu akurasi sebesar 0.9978252990213845, presisi 0.9977950634893669, recall 0.9978456350229938, dan f1-score sebesar 0.9978200798118901. Cell terakhir adalah proses menghitung jumlah data yang diprediksi dengan benar dan yang diprediksi dengan salah oleh model klasifikasi pada data uji. Dengan membandingkan prediksi yang dihasilkan oleh model dengan label yang sebenarnya dari data uji, dapat diidentifikasi berapa banyak prediksi yang tepat

dan prediksi yang tidak tepat. Pada penelitian ini, jumlah correct predictions adalah 19271, incorrect predictions 42, dan total data uji adalah 19313.

Setelah mengukur akurasi model Decision Tree, peneliti melanjutkan dengan melakukan cross-validation untuk mengevaluasi konsistensi dan kemampuan generalisasi model. Cross-validation dilakukan dengan membagi data menjadi beberapa subset dan menjalankan pelatihan serta pengujian secara bergantian pada setiap subset. Hasil cross-validation dengan 5 subset menunjukkan akurasi rata-rata sebesar 99.74%, dengan skor antar fold yang bervariasi antara 99.64% dan 99.88%. Ini mengindikasikan bahwa model memiliki kinerja yang stabil dan mampu memprediksi data baru dengan baik, serta tidak mengalami overfitting yang signifikan.

Tabel 2. Hasil Perbandingan Akurasi

Algoritma	Akurasi	Presisi	Standar Deviasi
Decision Tree	99.78%	99.77%	-
Decision Tree dan Cross Validation	99.74%	99.74%	0.0008

Tabel 2 di atas memperlihatkan hasil perbandingan akurasi dan presisi antara penggunaan algoritma Decision Tree dengan dan tanpa penerapan cross-validation. Dua metrik utama yang ditampilkan adalah akurasi dan presisi, dengan tambahan metrik standar deviasi untuk mengukur variasi hasil cross-validation.

Decision Tree Tanpa cross-validation menunjukkan akurasi sebesar 99.78% dan presisi sebesar 99.77%. Angka ini menunjukkan performa yang sangat tinggi dan konsisten dalam memprediksi churn pelanggan. Sedangkan Decision Tree dengan Cross Validation, didapatkan hasil akurasi rata-rata yang dicapai adalah 99.74% dengan presisi 99.74%, dan standar deviasi sebesar 0.0008. Standar deviasi yang sangat kecil menunjukkan bahwa hasil akurasi pada setiap fold cross-validation sangat konsisten dan hampir tidak ada variasi signifikan antara fold yang satu dengan yang lain. Dengan demikian, tabel ini mengonfirmasi bahwa algoritma decision tree, baik tanpa maupun dengan penerapan cross-validation, menunjukkan performa yang sangat baik dan konsisten dalam prediksi churn pelanggan. Implementasi cross-validation lebih lanjut memperkuat bukti bahwa model ini mampu menghasilkan prediksi yang stabil dan handal, bahkan ketika diuji pada subset data yang berbeda.

#### IV. KESIMPULAN

Dari pemrosesan yang telah dilakukan menggunakan algoritma *decision tree* untuk memprediksi *customer churn*, diperoleh hasil akurasi mencapai 99,7% dan presisi yang juga mencapai 99,7%. Dapat disimpulkan, pemodelan *decision tree* yang digunakan untuk memprediksi *customer churn* memiliki akurasi yang lebih tinggi dari algoritma *naive bayes* dalam

penelitian sebelumnya, yang akurasi mencapai 80%. Selain itu, hasil cross-validation menunjukkan konsistensi tinggi dengan rata-rata akurasi yang mendekati hasil pengujian langsung. Untuk penelitian berikutnya, disarankan untuk menggunakan algoritma lain seperti *Random Forest* atau *Gradient Boosting* untuk melihat apakah bisa meningkatkan akurasi.

## DAFTAR PUSTAKA

- [1] C. Olivia and Y. Sibaroni, "ANALISIS PREDIKSI CHURN MENGGUNAKAN METODE LOGISTIC REGRESSION DAN ALGORITMA DECISION TREE," Bandung, Aug. 2015. Accessed: May 31, 2024. [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/3020>
- [2] S. DI PENDUKUNG KEPUTUSAN PENERIMA DANA DESA DENGAN MENGGUNAKAN METODE PROMETHEE KECAMATAN GODEAN KABUPATEN SLEMAN Arif Budiman *et al.*, "PREDIKSI CUSTOMER CHURN PERUSAHAAN TELEKOMUNIKASI MENGGUNAKAN NAÏVE BAYES DAN K-NEAREST NEIGHBOR," 2019, [Online]. Available: <http://e-journal.janabadra.ac.id/>
- [3] Ni Wayan Wardani, Gede Rasben Dantes, and Gede Indrawan, "PREDIKSI CUSTOMER CHURN DENGAN ALGORITMA DECISION TREE C4.5 BERDASARKAN SEGMENTASI PELANGGAN PADA PERUSAHAAN RETAIL," *JURNAL RESISTOR*, vol. 1, pp. 16–24, Apr. 2018, doi: <https://doi.org/10.31598/jurnalresistor.v1i1.219>.
- [4] A. Nurzahputra, A. Ratna Safitri, and M. Aziz Muslim, "Klasifikasi Pelanggan pada Customer Churn Prediction Menggunakan Decision Tree," 2016. Accessed: May 31, 2024. [Online]. Available: <https://journal.unnes.ac.id/sju/prisma/article/view/21528/10288>
- [5] A. Sulistyowibowo, "Analisis Churn Nasabah Bank Dengan Pendekatan Machine Learning dan Pengelompokan Profil Nasabah dengan Pendekatan Clustering," vol. 2, no. 1, pp. 30–41, 2024, doi: 10.61132/konstruksi.v2i1.43.
- [6] Miryam Clementine and Arum, "Prediksi Churn Nasabah Bank Menggunakan Klasifikasi Naïve Bayes dan ID3," *Jurnal Processor*, vol. 17, no. 1, pp. 9–18, May 2022, doi: 10.33998/processor.2022.17.1.1170.
- [7] C. Mulia *et al.*, "Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Bank Customer Churn Menggunakan Algoritma Naïve bayes dan Logistic Regression," Jakarta Selatan, Aug. 2023. Accessed: May 31, 2024. [Online]. Available: <https://conference.upnvj.ac.id/index.php/senamika/article/view/2590>
- [8] R. Novendri, R. Andreswari, and O. N. Pratiwi, "IMPLEMENTASI DATA MINING UNTUK MEMPREDIKSI CUSTOMER CHURN MENGGUNAKAN ALGORITMA NAIVE BAYES IMPLEMENTATION OF DATA MINING TO PREDICT CUSTOMER CHURNS USING NAIVE BAYES ALGORITHM," Apr. 2021.
- [9] R. Alfarez, V. Purwayoga, J. Siliwangi No, K. Kahuripan, K. Tawang, and K. Tasikmalaya, "PENERAPAN NAÏVE BAYES UNTUK PREDIKSI CUSTOMER CHURN (STUDI KASUS: PT HUTCHISON 3 INDONESIA)," *Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI)*, vol. 05, 2024, Accessed: May 30, 2024. [Online]. Available: <https://jim.unindra.ac.id/index.php/jrami/article/view/8556>
- [10] Y. Christian, J. Putra, A. Winata, N. Ricky, R. Jeonanto, and H. Artikel, "PREDIKSI KUALITAS AIR MENGGUNAKAN ALGORITMA NAÏVE BAYES DAN RANDOM FOREST," Oct. 2022. doi: 10.24269/jkt.v6i2.1313.
- [11] A. M. Husein and M. Harahap, "Pendekatan Data Science untuk Menemukan Churn Pelanggan pada Sector Perbankan dengan Machine Learning," *Data Sciences Indonesia (DSI)*, vol. 1, no. 1, pp. 8–13, May 2021, doi: 10.47709/dsi.v1i1.1169.

- [12] A. Yulianto and Firmansyah, "Prediksi Customer Churn Pada Bisnis Retail Menggunakan Algoritma Naïve Bayes," *Riset dan E-Jurnal Manajemen Informatika Komputer*, vol. 6, no. 1, Oct. 2021, doi: 10.33395/remik.v4i1.11196.
- [13] N. Febriana, F. Fadzira, M. A. Senubekti, and R. Suharsih, "Prediksi Penilaian Kinerja Hakim Dengan Penerapan Machine Learning Menggunakan Tools Python," vol. 2, no. 1, 2024.
- [14] AZEEM and MUHAMMAD SHAHID, "Customer Churn Dataset," Kaggle. Accessed: Jun. 09, 2024. [Online]. Available: <https://www.kaggle.com/datasets/muhammadshahidazeem/customer-churn-dataset/data>
- [15] C. M. Sitorus, A. Rizal, and M. Jajuli, "Prediksi Risiko Perjalanan Transportasi Online Dari Data Telematik Menggunakan Algoritma Support Vector Machine," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, Aug. 2020, doi: 10.28932/jutisi.v6i2.2672.