Klasifikasi *Customer Relationship Management* Perusahaan Telekomunikasi Seluler Dengan Metode *Machine Learning*

[1]Muhammad Dahlan Kurnia, [2]Tukiyat, [3]Miswan

[1,3)]Mahasiswa Program Studi Magister Teknik Informatika, Universitas Pamulang, Tangerang, Banten
[2)]Dosen Universitas Pamulang/Perekayasa BRIN, Tangerang, Banten
Email: [1]emdeka1001@gmail.com, [2]dosen02711@unpam.ac.id, [3]wanwanvm@gmail.com

ABSTRACT

Customer Relationship Management (CRM) as the part of Enterprise Resource Planning (ERP) focuses on the company relationship with the costumer must be optimized. By the increasingly of tight business competition, CRM the cellular telecommunication company must be able to carry out the classification of loyal costumer and moving costumer (churn) as the initial steps to maintain costumer. This research aims to classification the numbers of costumers who are churn from the company to another company. The research approach was conducted by quantitative methods using secondary data. The data research was obtained from the data source of telco_dataset.csv. The data input was conducted by Python software. The results of the data collection are processed through the machine learning by decision tree, logistic regression and random forest methods. The results of research show the classifications by the methods of decision tree 78, 27%, logistic regression 78, 67% and random forest 78, 13% has the accuracy. Then, the classification model that suitable can be used to classify Customer Relationship Management. The cellular telecommunication company is the logistic regression method, because has more the high accuracy level. As the factors that have sensitive influential in classification of establishment that are the total of cost per month are issued by customers, the total of overall cost during being customers and the how long customers subscribe is.

Keywords: churn; Customer Relationship Management; decision tree; logistic regression; random forest.

ABSTRAK

Customer Relationship Management (CRM) sebagai bagian dari *Enterprise Resources Planning* (ERP) yang fokus pada hubungan perusahaan dengan pelanggan harus terus dioptimalkan. Dengan persaingan bisnis yang semakin ketat, CRM perusahaan telekomunikasi seluler harus mampu melakukan klasifikasi pelanggan setia dan pelanggan yang pindah (*churn*) sebagai langkah awal mempertahankan pelanggan. Penelitian ini bertujuan untuk mengkasifikasikan jumlah pelanggan yang *churn* dari perusahaan ke perusahaan lainnnya. Pendekatan penelitian dilakukan dengan metode kuantitatif dengan menggunakan data sekunder. Data penelitin diperoleh dari sumber data telco_dataset.csv. Input data dilakukan dengan *software* Python. Hasil pengumpulan data diolah melalui *machine learning* dengan metode *decision tree, logistic regression* dan *random forest*. Hasil penelitian menunjukkan bahwa klasifikasi dengan metode *decision tree* memiliki akurasi 78,27%, *logistic regression* 78,67% dan *random forest* 78,13%. Maka model klasifikasi yang cocok dapat digunakan untuk mengklasifikasikan *Customer Relationship Management* Perusahaan Telekomunikasi Seluler adalah metode *logistic regression*, karena mempunyai tingkat akurasi yang lebih tinggi. Adapun Faktor-faktor yang berpengaruh sensitif dalam pemodelan klasifikasi adalah total biaya per bulan yang dikeluarkan *customer*, total biaya keseluruhan selama menjadi *customer* dan lama *customer* berlangganan.

Kata kunci: *churn; Customer Relationship Management; decision tree; logistic regression; random forest.*

1. PENDAHULUAN

Customer Relationship Management (CRM) sebagai bagian dari *Enterprise Resources Planning* (ERP) yang fokus pada hubungan perusahaan dengan pelanggan harus terus dioptimalkan. Pelanggan merupakan salah satu faktor penentu sebuah perusahaan untuk terus bersaing dalam bisnis. Hal ini dilakukan agar pelanggan menjadi loyal terhadap perusahaan. Tingkat loyalitas ini dapat tercapai dengan adanya CRM yang tepat sasaran sehingga tercipta hubungan yang baik dan saling menguntungkan dengan para pelanggannya [1].

Perusahaan telekomunikasi seluler telah menjadi bisnis fundamental yang terus berkembang. Perusahaan telekomunikasi seluler baru bermunculan, secara langsung menambah kompetitor bagi perusahaan lama. Pelanggan dapat dengan mudah menggunakan haknya, untuk berpindah penyedia layanan dari satu operator ke operator lainnya. Banyaknya operator seluler mendorong persaingan bisnis yang semakin ketat [2].

Pada sisi lain, perkembangan dunia teknologi memicu perkembangan ilmu data mining. Data mining adalah serangkaian proses untuk menggali nilai tambah berupa informasi yang selama ini tidak diketahui secara manual dari suatu basis data. Informasi yang dihasilkan dapat diperoleh dengan cara mengekstraksi dan mengenali pola yang penting atau menarik dari data yang terdapat dalam basis data [3]. Salah satu konsen data mining adalah klasifikasi. Klasifikasi adalah proses mencari model dengan cara mengategorisasikan sehingga dapat memprediksi kelas yang tidak berlabel [4]. Salah satu tujuan klasifikasi adalah untuk meningkatkan kehandalan hasil yang diperoleh dari data [3].

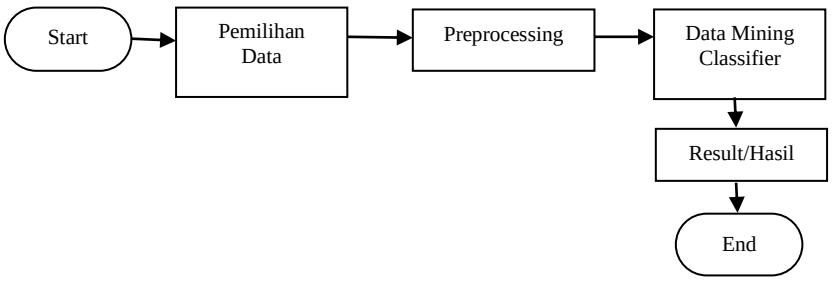
Dengan persaingan bisnis yang semakin ketat, CRM perusahaan telekomunikasi seluler harus mampu melakukan klasifikasi pelanggan setia dan pelanggan yang pindah (*churn*) sebagai langkah awal mempertahankan pelanggan. Karena ongkos mempertahankan pelanggan 5 kali lebih murah daripada memperoleh pelanggan baru, maka mempertahankan pelanggan dan mengurangi *churn rate* merupakan cara yang paling tepat bagi PT Telekomunikasi Seluler untuk mempertahankan *market share*-nya [5].

Dengan mempertimbangkan jumlah data yang ada, tujuan penelitian ini adalah membentuk model *data testing* yang dapat melakukan klasifikasi pelanggan pada perusahaan telekomunikasi seluler menggunakan metode data mining *decision tree, logistic regression* dan *random forest*, sehingga bisa dijadikan dasar pengambilan keputusan lebih lanjut terkait pelanggan yang akan pindah. Penerapan tiga metode data mining tersebut juga menjadi kebaruan dalam penelitian ini, dimana sebelumnya sudah banyak penelitian tentang klasifikasi pelanggan perusahaan telekomunikasi seluler menggunakan teknik klasifikasi data mining menggunakan metode yang lain. Seperti yang dilakukan oleh Iqbal Muhammad Latief dalam penelitian yang berjudul Prediksi Tingkat Pelanggan *Churn* Pada Perusahaan Telekomunikasi Dengan Algoritma *Adaboost* yang menyimpulkan bahwa algoritma *adaboost* dapat memprediksi masalah *churn* lebih baik dari algoritma *random forest* dan *xgboost* serta "*TotalCharges*" adalah fitur yang paling penting dalam memprediksi *churn* dengan tingkat akurasi 80% dari pada penelitian sebelumnya dengan algoritma *random forest* [2].

2. METODE

2.1. Rancangan Penelitian

Penelitian ini termasuk dalam jenis penelitian eksplorasi untuk membangun model klasifikasi CRM pelanggan *Churn*. Alur pikir dalam rancangan penelitian dapat ditunjukkan pada gambar 1.



Gambar 1. Alur Prosedur Penelitian

2.2. Data Penelitian

Penelitian ini menggunakan data sekunder dari data set pelanggan di sebuah perusahaan telekomunikasi seluler. Data penelitian dikompilasi dalam file telco_dataset.csv diunduh dari https://github.com/hafizmrf3/CustomerChurnPrediction/blob/main/telco_dataset.csv. Data penelitian yang diperoleh dengan ukuran file sebesar 706 KB. Data penelitian memiliki 6.950 entri (row) dan 13 fitur/atribut (column) yang akan digunakan dalam proses klasifikasi pelanggan. Berikut deskripsi 13 fitur data set.

Tabel 1. Deskripsi Data Penelitian

Fitur	Tipe Data	Deskripsi
UpdateAt	Date	Tanggal data dicatat
CustomerID	Numerik	ID unik customer
Gender	Kategorikal	Jenis kelamin customer (Male, Female)
SeniorCitizen	Kategorikal	Apakah ada customer lebih dari 60 tahun (Yes, No)
Partner	Kategorikal	Apakah customer memiliki pasangan (Yes, No)
Tenure	Numerik	Lama customer berlangganan (Bulan)
PhoneService	Kategorikal	Apakah customer menggunakan layanan telepon (Yes, No)
InternetService	Kategorikal	Apakah customer menggunakan layanan internet service provider (Yes, No)
StreamingTV	Kategorikal	Apakah customer menggunakan layanan streaming TV (Yes, No)
PaperlessBilling	Kategorikal	Apakah customer membayar dengan *e-billing/e-payment* (Yes, No)
MonthlyCharges	Numerik	Total biaya per bulan yang dikeluarkan customer
TotalCharges	Numerik	Total biaya keseluruhan customer
Churn	Kategorikal	Apakah customer churn (berpotensi pindah ke perusahaan lain) (Yes, No)

2.3. Preprocessing Data

Langkah berikutnya setelah data penelitian terkumpul adalah melakukan *preprocessing data*. *Preprocessing* merupakan pembersihan data yang *missing value*, menghilangkan data yang tidak digunakan dan normaliasi dalam perhitungan [6]. Berikut langkah *preprocessing data* yang digunakan dalam penelitian ini:

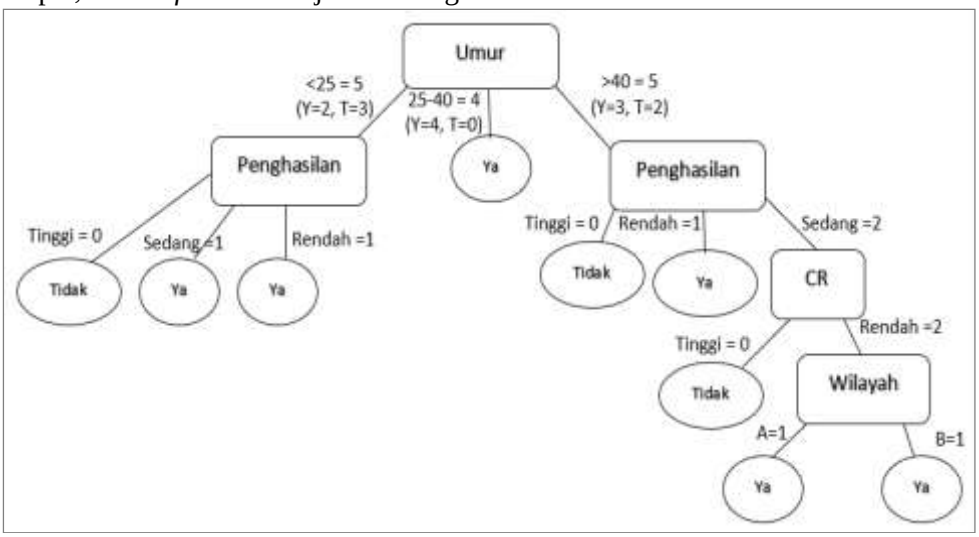
1) Penyesuaian data: merupakan proses menyesuaikan data pada setiap target. Data yang digunakan untuk melatih model memiliki pengaruh besar pada performa yang dapat dicapai. Fitur yang tidak relevan dapat berdampak negatif pada performa model. Seleksi fitur dilakukan dengan menggunakan berbagai macam metode seperti metode *filtering* dengan *Pearson Correlation* [7].
2) Pemisahan data: merupakan proses pemisahan data menjadi *data training* dan *data testing*.
3) Tranformasi data: merupakan proses mengubah data yang dipilih, sehingga sesuai untuk proses data mining. Proses ini merupakan proses kreatif dan sangat bergantung pada jenis atau pola informasi yang akan dicari dalam data penelitian. Dalam proses perubahan data akan lebih cepat menggunakan metode *One Hot Encoding* [2], yaitu mengubah nilai data menjadi fitur dan mengisinya dengan nilai kategorikal.
4) Normalisasi data: merupakan menyesuaikan isi data antar fitur agar bisa dan mudah digunakan dalam proses perhitungan metode data mining.

2.4. Data Mining

Data mining merupakan gabungan dari berbagai bidang ilmu, antara lain basis data, information retrieval, statistika, algoritma dan machine learning. Bidang ini telah berkembang sejak lama namun makin terasa pentingnya sekarang ini dimana muncul keperluan untuk mendapatkan informasi yang terkumpul selama bertahun - tahun. Data mining adalah cara menemukan informasi tersembunyi dalam sebuah basis data dan merupakan bagian dari proses *Knowledge Discovery in Databases* (KDD) untuk menemukan informasi dan pola yang berguna dalam data [8]. Ada banyak metode dalam data mining.

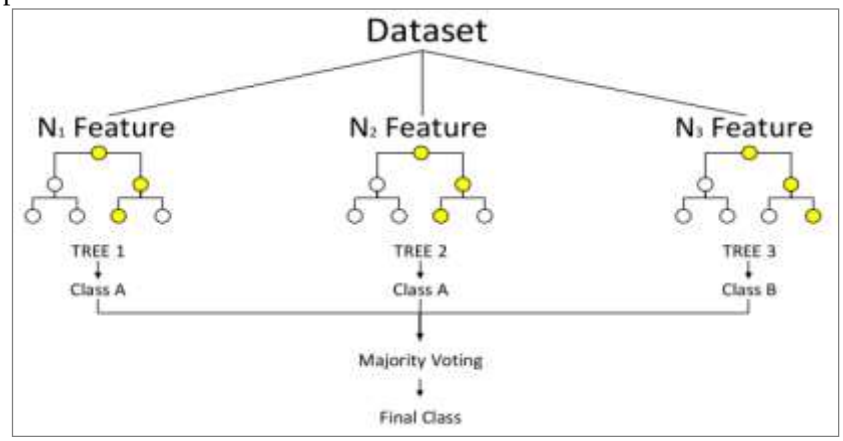
Penelitian ini menggunakan metode *decision tree, logistic regression* dan *random forest*. Ketiga metode ini merupakan metode khusus menyelesaikan masalah klasifikasi dengan teknik *supervised learning*. *Supervised learning* adalah teknik melatih mesin menggunakan data yang diberi label. Maksud dari pembelajaran yang diawasi adalah data label atau target ikut berperan sebagai 'supervisor' atau 'guru' yang mengawasi proses pembelajaran dalam mencapai tingkat akurasi atau presisi tertentu [7].

1) *Decision Tree*: adalah suatu *flowchart* seperti struktur pohon, dimana tiap titik internalnya (*internal node*) menunjukkan suatu test pada suatu atribut, tiap cabang (*branch*) merepresentasikan hasil dari test tersebut, dan *leaf node* menunjukkan kelas-kelas atau distribusi kelas [5]. Suatu *decision tree* yang terlihat seperti pada Gambar 2 merupakan *decision tree* yang merepresentasikan konsep pembelian mobil, yaitu untuk memprediksi apakah seorang pelanggan akan membeli mobil atau tidak. *Internal node* ditunjukkan dengan segi empat, dan *leaf node* ditunjukkan dengan oval.



Gambar 2. *Decision Tree* Pembelian Mobil

2) *Logistic Regression*: Model logistik regresi adalah suatu model statistik yang digunakan untuk mengetahui pengaruh variabel prediktor (X) terhadap variabel respon (Y) dengan variabel respon berupa data dikotomi yaitu bernilai 1 menyatakan bahwa variabel respon memiliki kriteria yang ditentukan dan 0 menyatakan bahwa variabel respon tidak memiliki kriteria yang ditentukan.
3) *Random Forest*: adalah metode pembelajaran ensemble untuk klasifikasi atau regresi yang beroperasi menggunakan menciptakan banyak pohon keputusan selama proses training dan menaruh hasil berupa mode kelas (klasifikasi) atau prediksi rata-rata (regresi) pohon individu [9]. Sama dengan *decision tree* pada proses deteksi dapat menggunakan formula *entropy* (*information gain*) dan *gini* (*impurity*). Dimana proses deteksi berlangsung dalam bentuk pohon keputusan ke bawah.



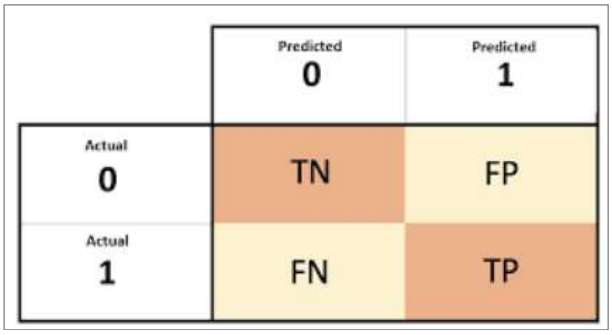
Gambar 3. Split *random forest*
(Sumber: [9], 2022)

2.5. Evaluasi

Tahapan mulai dari pemilihan data kemudian melakukan preprocessing yang didalamnya terdapat proses penyesuaian data, pemisahan data, transformasi data serta normalisasi data yang dihasilkan dari proses data mining, perlu ditampilkan dalam bentuk yang mudah dipahami oleh pihak yang berkepentingan. Pada tahap evaluasi, pengetahuan yang dihasilkan akan ditampilkan dalam bentuk nilai dari akurasi, presisi, *recall* yang terdapat dalam *confusion matrix* serta dalam bentuk nilai *area under curve* (AUC) yang terdapat dalam kurva *receiver operating characteristic* (ROC).

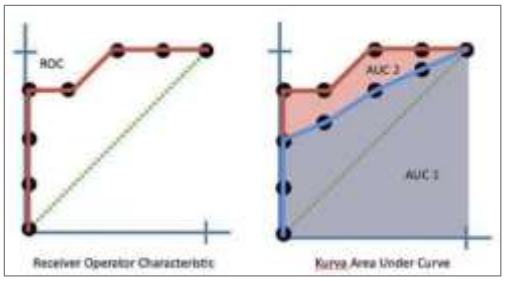
Ada 4 kondisi dalam *Confusion Matrix* [7]:

a. *True Positive* (TP): prediksi positif, real positif
b. *True Negative* (TN): prediksi negatif, real negatif
c. *False Positive* (FP): prediksi positif, real negatif
d. *False Negative* (FN): prediksi negatif, real positif



Gambar 4. *Confusion Matrix*
(Sumber: [7], 2021)

Pada *confusion matrix*, performa informasi hanya disajikan dalam bentuk angka. Untuk menampilkan informasi kinerja algoritma klasifikasi dalam bentuk grafik dapat digunakan kurva ROC.



Gambar 5. Kurva ROC
(Sumber: [7], 2021)

2.6. Python

Untuk mempercepat dan mempermudah serta menghilangkan kesalahan dalam proses perhitungan, maka penelitian ini memerlukan alat bantu pengolahan data. Ada banyak alat bantu pengolahan data saat ini, seperti SPSS, Rapid Miner, Orange, Python dan sebagainya. Alat bantu pengolahan data yang digunakan dalam penelitian ini adalah Python. Python sendiri merupakan Bahasa pemrograman tingkat tinggi yang menggunakan *system interpreter*, seperti halnya PHP dan Matlab [10].

Program yang dikembangkan Python dapat diperasionalisasi pada hampir semua sistem operasi baik Windows, Linux, Mac OS, Unix, dan juga sistem operasi pada perangkat lunak berbasis mobile seperti Android atau IOS. Dukungan Komunitas, dengan menggunakan Pyhton memiliki dukungan komunitas yang sangat kuat, karena Python bersifat *opensource*. Dengan komunitas yang baik, mempermudah pengguna untuk saling berbagi, dan mengembangkan bahasa pemrograman Python menjadi bahasa yang handal [10].

3. HASIL

3.1. Deskripsi Data Penelitian

Objek yang digunakan dalam penelitian ini adalah data pelanggan pada perusahaan telekomunikasi yang dapat di unduh di https://github.com/hafizmrf3/CustomerChurnPrediction/blob/main/telco_dataset.csv. Setelah file .csv sebagai data penelitian siap, maka akan dibaca oleh python dan diubah menjadi *data frame*, setelah itu bisa ditampilkan.

```
In [48]: import pandas as pd
         df1 = pd.read_csv('telco_dataset.csv', sep=',')

In [49]: df1.head(10)
```

Out[49]:

atedAt	customerID	gender	SeniorCitizen	Partner	tenure	PhoneService	StreamingTV	InternetService	PaperlessBilling	MonthlyCharges	TotalCharges	Churn
202006	45759018157	Female	No	Yes	1	No	No	Yes	Yes	29.05	29.85	No
202006	45315483266	Male	No	Yes	60	Yes	No	No	Yes	20.50	1198.80	No
202006	45236961615	Male	No	No	5	Yes	Yes	Yes	No	104.10	541.90	Yes
202006	45929827382	Female	No	Yes	72	Yes	Yes	Yes	Yes	115.50	8312.75	No
202006	45305082233	Female	No	Yes	56	Yes	Yes	Yes	No	81.25	4620.40	No
202006	45072364214	Male	No	No	44	Yes	Yes	Yes	Yes	85.25	3704.15	No
202006	45410681487	Male	No	Yes	38	Yes	Yes	Yes	No	80.00	3182.95	Yes
202006	45131942507	Male	No	Yes	12	Yes	No	Yes	Yes	64.00	959.90	No
202006	45105270921	Female	No	Yes	71	Yes	No	No	No	19.90	1397.30	No
202006	45874886225	Male	No	Yes	19	Yes	No	No	No	19.90	357.70	No

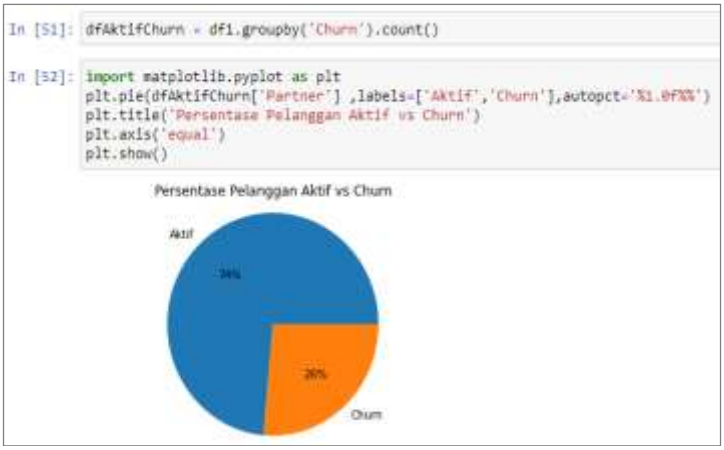
Gambar 6. Tampilan Data Penelitian

```
In [50]: df1.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6950 entries, 0 to 6949
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   UpdatedAt         6950 non-null   int64
 1   customerID        6950 non-null   int64
 2   gender            6950 non-null   object
 3   SeniorCitizen     6950 non-null   object
 4   Partner           6950 non-null   object
 5   tenure            6950 non-null   int64
 6   PhoneService      6950 non-null   object
 7   StreamingTV       6950 non-null   object
 8   InternetService   6950 non-null   object
 9   PaperlessBilling  6950 non-null   object
 10  MonthlyCharges    6950 non-null   float64
 11  TotalCharges      6950 non-null   float64
 12  Churn             6950 non-null   object
dtypes: float64(2), int64(3), object(8)
memory usage: 706.0+ KB
```

Gambar 7. Info Data Penelitian

Data penelitian terdiri dari 6.950 entri dimulai dari 0 sampai 6.949 dan terdiri dari 13 fitur di mulai dari 0 sampai 12.

Selanjutnya, sebelum dilakukan penyesuaian data perlu diketahui persentase jumlah pelanggan aktif dan pelanggan *churn*.



Gambar 8. Prosentase Pelanggan Aktif dan *Churn*

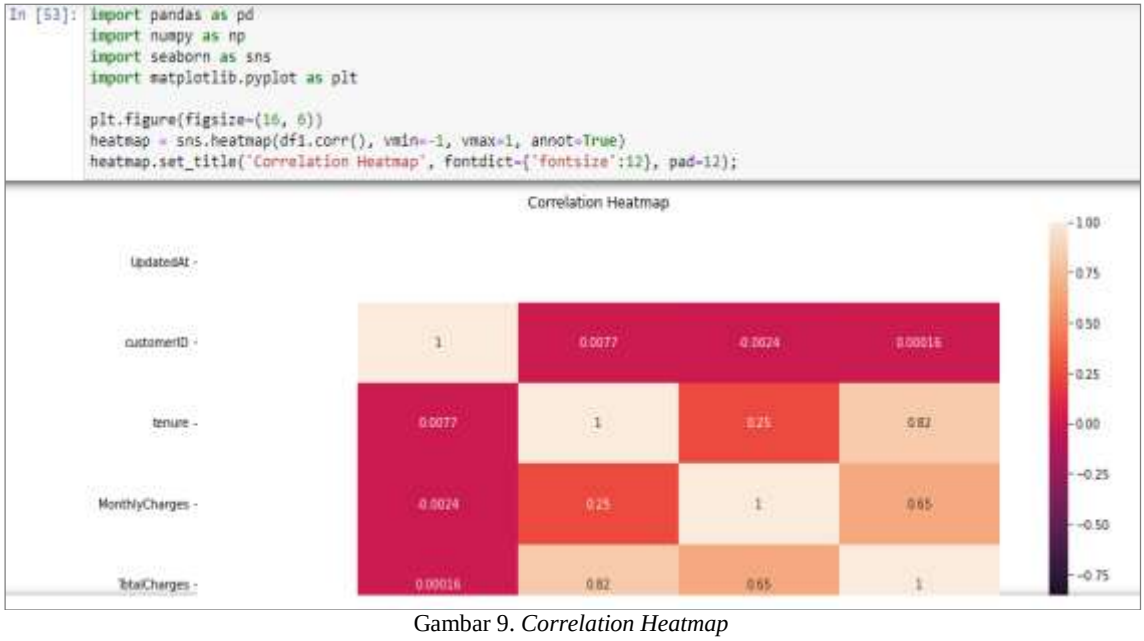
Jumlah pelanggan yang aktif lebih banyak jika dibandingkan dengan jumlah pelanggan yang *churn*. Selisihnya sekitar:

Jumlah pelanggan aktif : 74% x 6.950 pelanggan = 5.143 pelanggan
Jumlah pelanggan churn: 26% x 6.950 pelanggan = 1.807 pelanggan
Selisih pelangaan aktif dengan *churn*: 5.143 - 1.807 = 3.336 pelanggan

3.2. Pengolahan Data Penelitian

Setelah mengetahui persentase jumlah pelanggan, selanjutnya melakukan penyesuaian data. Penyesuaian data dilakukan dengan menghapus fitur yang tidak relevan terhadap penelitian. Proses seleksi fitur dihitung menggunakan metode *filtering pearson correlation* yang berfungsi untuk mengukur kedekatan hubungan antara dua fitur menggunakan suatu garis lurus. Koefisien *pearson correlation* memiliki jangkauan dari angka -1 hingga +1. Jika angka nol berarti tidak ada sama sekali hubungan berdasar garis lurus antara kedua fitur, dan jika angka mendekati -1 atau +1 berarti kedua fitur memiliki hubungan garis lurus yang hampir sempurna. Dalam proses penyesuaian data mesti menghilangkan hubungan garis lurus dua fitur yang nyaris sempurna atau sama sekali tidak ada hubungan, jika terjadi maka salah satu fitur harus dihapus.

Selanjutnya agar lebih mudah melihat hubungan dua fitur maka ditampilkan dalam bentuk *heatmap*. Hubungan garis lurus dua fitur yang memiliki angka *pearson correlation* tinggi atau sama sekali tidak ada hubungan (0) akan berwarna lebih terang.



Gambar 9. *Correlation Heatmap*

Tampak bahwa fitur "UpdateAt" dan "customerID" berwarna putih terang. Hal ini menggambarkan kedua fitur tidak memiliki keterkaitan sama sekali dengan fitur lain. Maka kedua fitur ini perlu dihapus.

```
In [54]: df1.drop('customerID', axis=1, inplace=True)
         df1.drop('UpdatedAt', axis=1, inplace=True)
         df1.head(10)
```

Out[54]:

	gender	SeniorCitizen	Partner	tenure	PhoneService	StreamingTV	InternetService	PaperlessBilling	MonthlyCharges	TotalCharges	Churn
0	Female	No	Yes	1	No	No	Yes	Yes	29.85	29.85	No
1	Male	No	Yes	60	Yes	No	No	Yes	20.50	1198.80	No
2	Male	No	No	5	Yes	Yes	Yes	No	104.10	541.90	Yes
3	Female	No	Yes	72	Yes	Yes	Yes	Yes	115.50	8312.75	No
4	Female	No	Yes	56	Yes	Yes	Yes	No	81.25	4620.40	No
5	Male	No	No	44	Yes	Yes	Yes	Yes	85.25	3704.15	No
6	Male	No	Yes	39	Yes	Yes	Yes	No	80.00	3182.95	Yes
7	Male	No	Yes	12	Yes	No	Yes	Yes	84.60	959.90	No
8	Female	No	Yes	71	Yes	No	No	No	19.90	1397.30	No
9	Male	No	Yes	19	Yes	No	No	No	19.90	357.70	No

Gambar 10. Data penelitian teratas setelah fitur UpdateAt dan customerID dihapus

Selanjutnya banyak atribut memiliki nilai kategorikal berbentuk *text*, seperti fitur "gender" memiliki nilai "Female" atau "Male", fitur "Partner" memiliki nilai "Yes" atau "No" yang perlu diubah menjadi 0 atau 1. Misalnya fitur "Partner" bernilai "Yes"=1 atau "No"=0. Maka cara paling mudah melakukan perubahan ini dengan metode *one hot encoding*.

```
In [55]: df1 = pd.get_dummies(df1, columns=["gender"])
         df1 = pd.get_dummies(df1, columns=["SeniorCitizen"])
         df1 = pd.get_dummies(df1, columns=["Partner"])
         df1 = pd.get_dummies(df1, columns=["PhoneService"])
         df1 = pd.get_dummies(df1, columns=["StreamingTV"])
         df1 = pd.get_dummies(df1, columns=["InternetService"])
         df1 = pd.get_dummies(df1, columns=["PaperlessBilling"])
         df1 = pd.get_dummies(df1, columns=["Churn"])
```

Gambar 11. Metode *One Hot Encoding* dengan Fungsi *get_dummies( )*

```
In [56]: df1.head(5)
```

Out[56]:

	tenure	MonthlyCharges	TotalCharges	gender_Female	gender_Male	SeniorCitizen_No	SeniorCitizen_Yes	Partner_No	Partner_Yes
0	1	29.85	29.85	1	0	1	0	0	1
1	60	20.50	1198.80	0	1	1	0	0	1
2	5	104.10	541.90	0	1	1	0	1	0
3	72	115.50	8312.75	1	0	1	0	0	1
4	56	81.25	4620.40	1	0	1	0	0	1

Gambar 12. Data Penelitian Teratas Lembar ke-1

```
In [56]: df1.head(5)
```

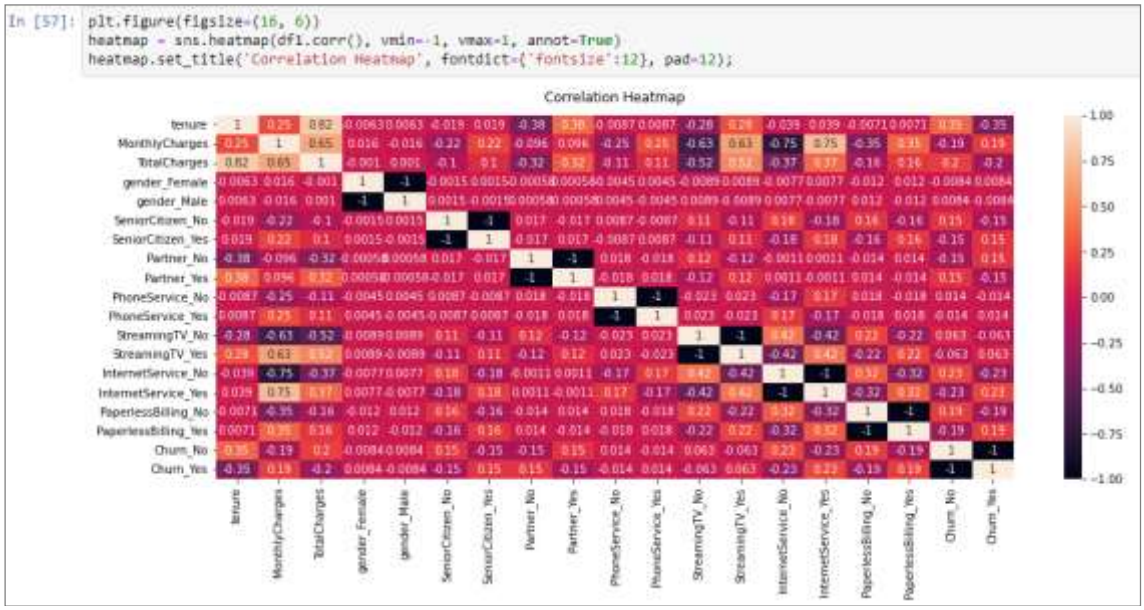
Out[56]:

_No	PhoneService_Yes	StreamingTV_No	StreamingTV_Yes	InternetService_No	InternetService_Yes	PaperlessBilling_No	PaperlessBilling_Yes	Churn_No	Churn_
1	0	1	0	0	1	0	1	1	
0	1	1	0	1	0	0	1	1	
0	1	0	1	0	1	1	0	0	
0	1	0	1	0	1	0	1	1	
0	1	0	1	0	1	1	0	1	

Gambar 13. Data Penelitian Teratas Lembar ke-2

Maka hasil dari penerapan metode *one hot encoding* yaitu fitur "gender" langsung terbagi menjadi dua fitur, yaitu "gender_Female" dan "gender_Male" dengan nilai 1 dan 0. Hal ini juga terjadi pada fitur lain yang diberikan fungsi yang sama. Selanjutnya perlu melakukan *fitur selection* dengan

metode *filtering pearson correlation* untuk yang kedua kali. Hal ini disebabkan proses *one hot encoding* yang memunculkan fitur-fitur baru.



Gambar 14. *Correlation Heatmap* Kedua

Beberapa fitur tampak berada pada nilai +1 atau -1. Contoh "gender_Female" dan "gender_Male" memiliki nilai +1 dan -1 berarti kedua fitur ini memiliki hubungan yang nyaris sempurna. Maka salah satunya perlu dihapus. Hal ini juga perlu dilakukan untuk fitur-fitur lain dengan kasus yang sama.

```
In [58]: df1.drop('gender_Female', axis=1, inplace=True)
         df1.drop('SeniorCitizen_No', axis=1, inplace=True)
         df1.drop('Partner_No', axis=1, inplace=True)
         df1.drop('PhoneService_No', axis=1, inplace=True)
         df1.drop('StreamingTV_No', axis=1, inplace=True)
         df1.drop('InternetService_No', axis=1, inplace=True)
         df1.drop('PaperlessBilling_No', axis=1, inplace=True)
         df1.drop('Churn_No', axis=1, inplace=True)
```

Gambar 15. Proses Penghapusan Fitur

Setelah fitur-fitur yang tidak relevan dihapus, selanjutnya perlu dilakukan pemisahan data. Dari data penelitian yang ada, dipisahkan sebanyak 80% untuk *data training* dan sisanya untuk *data testing*.

```
In [90]: import sklearn.model_selection as ms
         X_train, X_test, y_train, y_test = ms.train_test_split(X,y, test_size=0.8)
```

Gambar 16. Proses Pemisahan *Data Testing* dan *Data Training*

Selanjutnya "X_Train" akan berisi semua fitur dan "y_train" akan berisi *target fitur*, yang akan dipakai pada proses *training model*. Kemudian "X_test" dan "y_test" akan berisi data testing untuk mengukur kinerja model. Setelah proses pemisahan data, selanjutnya adalah proses normalisasi data. Normalisasi data sangat penting agar nilai terkecil dengan nilai terbesar memiliki jangkauan yang sama. Contoh pada fitur "TotalCharges" ada yang bernilai 29,5, ada juga yang bernilai 8.312,75. Dengan jangkauan terlalu jauh seperti ini sering kali menimbulkan error ketika proses perhitungan, maka perlu dinormalisasi.

```
In [91]: import sklearn.preprocessing as pp
         scl = pp.StandardScaler(copy=True, with_mean=True, with_std=True)
         scl.fit(X_train)
         X_train = scl.transform(X_train)
         X_test = scl.transform(X_test)
```

Gambar 17. Proses Normalisasi

4. PEMBAHASAN

Setelah dilakukan normalisasi terhadap data yang ada, langkah selanjutnya adalah mengukur kinerja model dari metode data mining yang digunakan, yaitu *decision tree, logistic regression dan random forest*. Tampak pada gambar berikut, hasil kinerja model baik itu akurasi, presisi maupun *recall* dari metode *decision tree*.

```
In [92]: import sklearn.tree as tree
         import sklearn.metrics as met
         model = tree.DecisionTreeClassifier(criterion='entropy', max_depth=5)
         model.fit(X_train, y_train)

         y_prediksi = model.predict(X_test)
         print(y_prediksi)
         score = met.accuracy_score(y_test, y_prediksi)
         print("accuracy Decision Tree =", score)
         precision = met.precision_score(y_test, y_prediksi)
         print("precision Decision Tree =", precision)
         recall = met.recall_score(y_test, y_prediksi)
         print("recall Decision Tree =", recall)
         auc = met.roc_auc_score(y_test, y_prediksi)
         print("AUC Decision Tree =",auc)

         [0 0 0 ... 0 0 0]
         accuracy Decision Tree = 0.7827338129496403
         precision Decision Tree = 0.6331828442437923
         recall Decision Tree = 0.38850415512465375
```

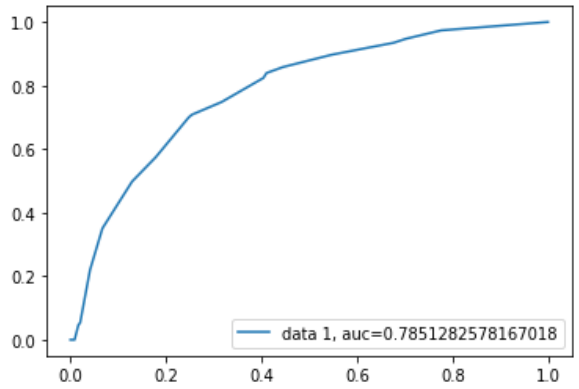
Gambar 18. Proses dan Hasil Pengukuran Kinerja Model *Decision Tree*

Karena hasil klasifikasi terdiri dari 2 kelas yaitu *churn* dan tidak *churn*, maka cukup kita perhatikan nilai akurasi. Tetapi perlu ditekankan bahwa yang ditindaklanjuti oleh perusahaan adalah kelas pelanggan *churn*. Hal ini disebabkan jumlah pelanggan *churn* lebih sedikit dibanding pelanggan setia/tidak *churn*.

Klasifikasi menggunakan metode *decision tree* menggunakan kriteria *entropy* dengan kedalaman maksimal *tree* 5, hasilnya dengan menggunakan data yang telah direduksi menggunakan fitur seleksi dari ekstraksi fitur sebanyak 80% x 6.950 = 5.560 data menggunakan *decision tree* yang mengklasifikasikan data pada data testing (20% x 6.950 = 1.390 data) dengan benar sebesar 78,27%. Sehingga klasifikasi dapat dengan tepat mengklasifikasikan data pelanggan *churn* pada *data testing*, dibanding harus memilih acak 50:50 atau 50%.

Adapun hasil kinerja model dari metode *decision tree* secara visual dapat dilihat melalui kurva ROC berikut ini, dengan melihat nilai AUC. Nilai AUC mendekati 1 berarti hasil kinerja model sudah cukup baik, tetapi jika mendekati atau sama dengan 0.5 berarti kurang baik.

```
In [23]: import matplotlib.pyplot as plt
         y_pred_proba = model.predict_proba(X_test)[::,1]
         fp, tp, _ = met.roc_curve(y_test, y_pred_proba)
         auc = met.roc_auc_score(y_test, y_pred_proba)
         plt.plot(fp,tp,label="data 1, auc="+str(auc))
         plt.legend(loc=4)
         plt.show()
```



Gambar 19. Visualisasi kurva ROC dari Metode *Decision Tree*

Selanjutnya tampak pada gambar berikut, hasil kinerja model baik itu akurasi, presisi maupun *recall* dari metode *logistic regression*.

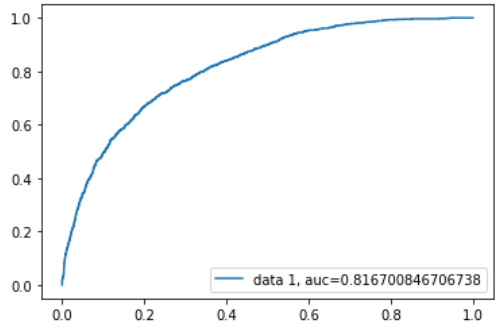
```
In [93]: import sklearn.linear_model as lm
         import sklearn.metrics as met
         model = lm.LogisticRegression(solver='lbfgs')
         model.fit(X_train, y_train)

         y_prediksi = model.predict(X_test)
         print(y_prediksi)
         score = met.accuracy_score(y_test, y_prediksi)
         print("accuracy Logistic Regression =", score)
         precision = met.precision_score(y_test, y_prediksi)
         print("precision Logistic Regression =", precision)
         recall = met.recall_score(y_test, y_prediksi)
         print("recall Logistic Regression =", recall)
         auc = met.roc_auc_score(y_test, y_prediksi)
         print("AUC Logistic Regression =",auc)

         [0 0 1 ... 0 0 0]
         accuracy Logistic Regression = 0.7866906474820143
         precision Logistic Regression = 0.6075
         recall Logistic Regression = 0.5048476454293629
```

Gambar 20. Proses dan Hasil Pengukuran Kinerja Model *Logistic Regression*

Klasifikasi menggunakan metode *logistic regression* terhadap data yang telah direduksi menggunakan fitur seleksi dari ekstraksi fitur sebanyak 80% x 6.950 = 5.560 data menggunakan *logistic regression* yang mengklasifikasikan data pada *data testing* (20% x 6.950 = 1.390 data) dengan benar sebesar 78,67%. Sehingga klasifikasi lumayan tepat mengklasifikasikan data pelanggan *churn* pada *data testing*, lebih tinggi 0,40% dibanding metode *decision tree*.

Adapun hasil kinerja model dari metode *logistic regression* secara visual dapat dilihat melalui kurva ROC berikut ini, dengan melihat nilai AUC.

```
In [26]: import matplotlib.pyplot as plt
         y_pred_proba = model.predict_proba(X_test)[::,1]
         fp, tp, _ = met.roc_curve(y_test, y_pred_proba)
         auc = met.roc_auc_score(y_test, y_pred_proba)
         plt.plot(fp,tp,label="data 1, auc="+str(auc))
         plt.legend(loc=4)
         plt.show()
```



Gambar 21. Visualisasi Kurva ROC dari Metode *Logistic Regression*

Selanjutnya tampak pada gambar berikut, hasil kinerja model baik itu akurasi, presisi maupun *recall* dari metode *random forest*.

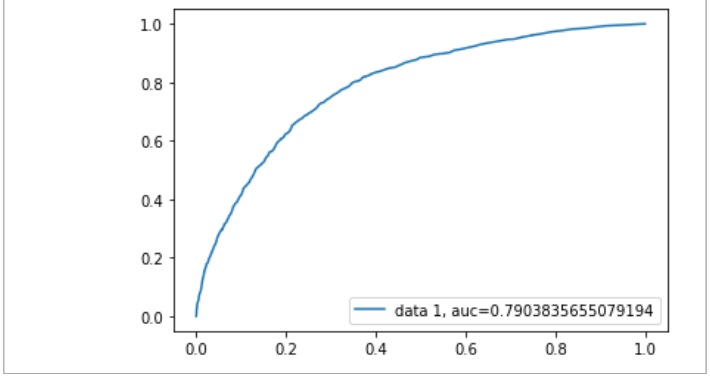
```
In [94]: import sklearn.ensemble as ens
         import sklearn.metrics as met
         model = ens.RandomForestClassifier(n_estimators=200, random_state=0)
         model.fit(X_train, y_train)

         y_prediksi = model.predict(X_test)
         print(y_prediksi)
         score = met.accuracy_score(y_test, y_prediksi)
         print("accuracy Random Forest =", score)
         precision = met.precision_score(y_test, y_prediksi)
         print("precision Random Forest =", precision)
         recall = met.recall_score(y_test, y_prediksi)
         print("recall Random Forest =", recall)
         auc = met.roc_auc_score(y_test, y_prediksi)
         print("AUC Random Forest =",auc)

         [0 0 0 ... 0 0 0]
         accuracy Random Forest = 0.781294964028777
         precision Random Forest = 0.5956375838926175
         recall Random Forest = 0.4916897506925208
```

Gambar 22. Proses dan Hasil Pengukuran Kinerja Model *Random Forest*

Klasifikasi menggunakan metode *random forest* terhadap data yang telah direduksi menggunakan fitur seleksi dari ekstraksi fitur sebanyak 80% x 6.950 = 5.560 data menggunakan *random forest* yang mengklasifikasikan data pada data testing (20% x 6.950 = 1.390 data) dengan benar sebesar 78,12%. Sehingga klasifikasi lumayan tepat mengklasifikasikan data pelanggan *churn* pada *data testing*, lebih rendah 0,15% dibanding metode *decision tree*.

Adapun hasil kinerja model dari metode *random forest* secara visual dapat dilihat melalui kurva ROC berikut ini, dengan melihat nilai AUC.

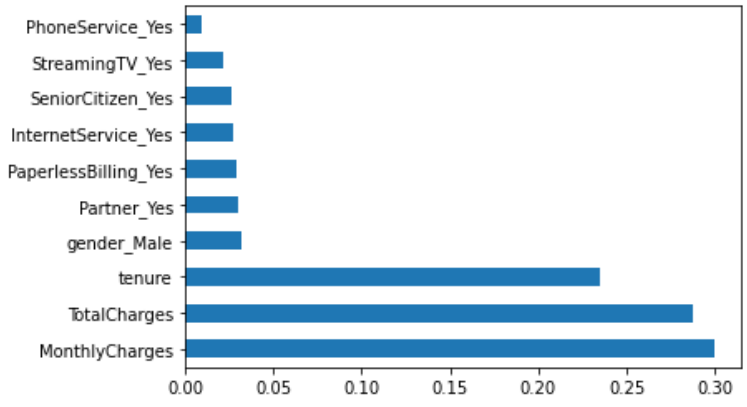
```
In [28]: import matplotlib.pyplot as plt
         y_pred_proba = model.predict_proba(X_test)[::,1]
         fp, tp, _ = met.roc_curve(y_test, y_pred_proba)
         auc = met.roc_auc_score(y_test, y_pred_proba)
         plt.plot(fp,tp,label="data 1, auc="+str(auc))
         plt.legend(loc=4)
         plt.show()
```



Gambar 22. Visualisasi kurva ROC dari metode *random forest*

Akhirnya model ini juga dapat mengetahui fitur penentu pelanggan yang akan melakukan *churn*. Berikut perintah dan hasil tampilan 10 fitur penentu yang mempengaruhi *churn* dengan membaca modul *feature_importances* di dalam model.

```
In [25]: import pandas as pd

         fitur_krusial = pd.Series(model.feature_importances_, index=X.columns)
         fitur_krusial.nlargest(10).plot(kind='barh')

Out[25]: <AxesSubplot: >
```



Gambar 23. Proses dan Hasil Fitur Penentu

5. KESIMPULAN

Setiap aplikasi ERP umumnya memiliki modul CRM. Modul CRM menghasilkan database pelanggan. Database pelanggan bisa dimanfaatkan dalam rangka analisis, prediksi, asosiasi atau konsen lain, yang berorientasi akhir pada peningkatan profitabilitas perusahaan. Berdasarkan hasil dan pembahasan pada bagian sebelumnya, dapat disimpulkan bahwa dengan memanfaatkan data penelitian telco_dataset.csv, metode *decision tree, logistic regression* dan *random forest* berhasil membentuk model *data testing* yang dapat melakukan klasifikasi pelanggan pada perusahaan telekomunikasi seluler dengan tingkat akurasi di atas 75%. Hasil penelitian menunjukkan bahwa klasifikasi dengan metode *decision tree* memiliki akurasi 78,27%, *logistic regression* 78,67% dan *random forest* 78,13%. Maka model klasifikasi yang paling cocok untuk digunakan dalam mengklasifikasikan *Customer Relationship Management* Perusahaan Telekomunikasi Seluler adalah metode *logistic regression*.

Selanjutnya juga disimpulkan fitur "Monthly Charges" (total biaya per bulan yang dikeluarkan customer), "TotalCharges" (total biaya keseluruhan selama menjadi customer) dan "tenure" (lama customer berlangganan) menjadi 3 faktor utama yang dapat menentukan pelanggan *churn* atau tetap setia terhadap produk/layanan.

Untuk penelitian selanjutnya, disarankan tetap fokus pada peningkatan angka akurasi, tentunya dengan menggunakan metode klasifikasi lain. Jika sudah mencapai nilai akurasi di atas 85%, disarankan untuk melakukan kombinasi metode data mining tersebut dengan metode *decision support system* seperti *weighted product* (WP) atau yang lainnya, sehingga dapat mengurangi biaya promosi terhadap pelanggan yang akan *churn*.

6. UCAPAN TERIMAKASIH

Terimakasih kepada:

1) Dr. Tukiyat, M.Si. atas bimbingan, kritik dan saran dalam penyusunan jurnal
2) Dr. Murni Handayani, M.Sc. atas informasi publikasi jurnal
3) Duta Arief Christianto dan Miswan, rekan mahasiswa satu angkatan di Magister Teknik Informatika Universitas angkatan tahun 2021
4) Isteri, anak-anak dan keluarga tercinta
5) Rekan kerja di Yayasan Hidayaturrohman Teluknaga

7. DAFTAR PUSTAKA

[1] F. Ardiansyah, F. Hamdan, S. Sugiyanto, and I. Wahyu Siadi, "Klasifikasi Customer Relationship Management Menggunakan Dataset KDD Cup 2009 dengan Teknik Reduksi Dimensi," *Komputika J. Sist. Komput.*, vol. 11, no. 2, pp. 193–202, 2022, doi: 10.34010/komputika.v11i2.6498.

[2] I. M. Latief, A. Subekti, and W. Gata, "Prediksi Tingkat Pelanggan Churn Pada Perusahaan Telekomunikasi Dengan Algoritma Adaboost," *J. Inform.*, vol. 21, no. 1, pp. 34–43, 2021, doi: 10.30873/ji.v21i1.2867.

[3] K. Khoirunnisa, L. Susanti, I. T. Rokhmah, and L. Stianingsih, "Prediksi Siswa Smk Al-Hidayah Yang Masuk Perguruan Tinggi Dengan Metode Klasifikasi," *J. Inform.*, vol. 8, no. 1, pp. 26–33, 2021, doi: 10.31294/ji.v8i1.9163.

[4] F. Reviantika, Y. Azhar, and G. I. Marthasari, "Analisis Klasifikasi SMS Spam Menggunakan Logistic Regression," *Repositor*, vol. 3, no. 4, pp. 387–392, 2021.

[5] R. Govindaraju, T. Simatupang, and T. A. Samadhi, "Perancangan Sistem Prediksi Churn Pelanggan Pt. Telekomunikasi Seluler Dengan Memanfaatkan Proses Data Mining," *J. Inform.*, vol. 9, no. 1, 2009, doi: 10.9744/informatika.9.1.33-42.

[6] A. Handayanto, K. Latifa, N. D. Saputro, and R. R. Waliansyah, "Analisis dan Penerapan Algoritma Support Vector Machine (SVM) dalam Data Mining untuk Menunjang Strategi Promosi," *JUITA J. Inform.*, vol. 7, no. 2, p. 71, 2019, doi: 10.30595/juita.v7i2.4378.

[7] K. Kristiawan and A. Widjaja, "Perbandingan Algoritma Machine Learning dalam Menilai Sebuah Lokasi Toko Ritel," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, pp. 35–46, 2021, doi: 10.28932/jutisi.v7i1.3182.

[8] L. P. Muri, B. Pramono, and J. Y. Sari, "Prediksi tingkat penyakit demam berdarah di kota kendari menggunakan metode," *semanTIK*, vol. 4, no. 1, pp. 103–112, 2018.

[9] N. Ghaniaviyanto Ramadhan, F. Dharma Adhinata, A. Jala, T. Segara, P. Rakhmadani, and F. Informatika, "Deteksi Berita Palsu Menggunakan Metode Random Forest dan Logistic Regression," *J. Ris. Komputer)*, vol. 9, no. 2, pp. 2407–389, 2022, doi: 10.30865/jurikom.v9i2.3979.

[10] E. Retnoningsih and R. Pramudita, "Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python," *Bina Insa. Ict J.*, vol. 7, no. 2, p. 156, 2020, doi: 10.51211/biict.v7i2.1422.