WEBSITE PREDIKSI CUSTOMER *CHURN* UNTUK MEMPERTAHANKAN PELANGGAN PADA PERUSAHAAN TELEKOMUNIKASI

Mohammad Amirulhaq Iskandar, Ulinnuha Latifa
Program Studi S1 Teknik Elektro, Fakultas Teknik
Universitas Singaperbangsa Karawang, Jl. HS. Ronggo Waluyo, Karawang, Indonesia
Mohammad.amirulhaq19072@student.unsika.ac.id

ABSTRAK

Fenomena churn, yaitu perilaku pelanggan yang beralih ke penyedia layanan telekomunikasi lain. Churn mengurangi keuntungan dan pendapatan perusahaan. Mempertahankan pelanggan yang ada menjadi tantangan utama, karena mendapatkan pelanggan baru lebih sulit dan biayanya lebih tinggi. Untuk mengatasi hal ini, perusahaan telekomunikasi perlu memprediksi kapan pelanggan akan churn. Prediksi customer *churn* dapat dilakukan dengan teknik Machine Learning. Dalam *Machine Learning*, diperlukan algoritma yang memiliki kemampuan untuk melakukan klasifikasi terhadap pelanggan apakah akan churn atau tidak. K-nearest neighbors (KNN) merupakan salah satu algoritma dari machine learning. Tahapan penelitian ini adalah dengan menerapkan *Artificial Intelegence proyek cycle* yaitu *problem scoping*, a*cquisition*, *data exploration*, *modelling*, dan *deployment*. Berdasarkan hasil evaluasi pengujian dan implementasi model tersebut menghasilkan akurasi 81% dan mendaptkan *goodfit*. Hasil dari model tersebut menunjukkan bahwa terdapat 817 prediksi yang benar dalam memprediksi pelanggan yang akan churn, namun terdapat 183 prediksi yang salah. Sementara itu, dalam memprediksi pelanggan yang tidak akan churn, terdapat 842 prediksi yang benar dan 203 prediksi yang salah. Dari hasil model tersebut diimplementasikan ke sebuah website yang memiliki berberapa fitur seperti halaman utama terdapat data nilai *churn* dan tidak *churn*, halaman memasukan data pelanggan, halaman data pelanggan *churn*, halaman pelanggan tidak *churn*. Website tersebut terhubung dengan database menggunakan server localhost. Website tersebut dapat digunakan untuk membantu selaku para perusahaan dibidang telekomunikasi untuk memprediksi customer *churn*.

Kata kunci: *KNN, Customer Churn, Website, Prediksi, Telekomunikasi*

1. PENDAHULUAN

Perkembangan teknologi yang semakin maju, menyebabkan kebutuhan akan informasi dan komunikasi semakin bertambah karenanya manusia menjadi lebih mudah dalam menyelesaikan masalah informasi dan juga komunikasi haya dengan menggunakan berbagai perangkat teknologi yang dimiliki atau yang ada. Perangkat teknologi sebagai penyebaran informasi dan komunikasi dapat ditemukan pada sebuah perusahaan telekomunikasi.

Perusahaan telekomunikasi merupakan menjadi salah satu penyedia layanan telekomunikasi seperti telepon dan juga akses komunikasi data. Tentunya layanan telekomunikasi pada sebuah perusahaan diperuntukan kepada pelanggan atau customer. Keberadaan pelanggan atau customer bagi sebuah perusahaan sangatlah penting.

Pelanggan sebagai pelaku konsumen yang akan menggunakan jasa atau produk yang ditawarkan dan dijual. Tanpa keberadaan pelanggan, sebuah perusahaan telekomunikasi tidak akan memiliki keuntungan yang menjadi tujuan utama dibangunnya sebuah bisnis.

Untuk mendapatkan hati pelanggan setiap perusahaan pastinya memiliki cara tersendiri dalam menawarkan layanannya yang berkualitas agar dapat menarik pelanggan sebanyak mungkin. Namun pada sebuah perusahaan telekomunikasi terdapat sebuah fenomena yaitu *churn*.

Churn adalah cara para pelanggan memutuskan untuk pindah dari satu perusahaan ke perusahaan yang lain karena ada penawaran yang lebih menarik. Fenomena *churn* ini tentu saja merugikan, karena mengurangi keuntungan atau pendapatan perusahaan [1].

Mempertahankan pelanggan yang ada menjadi isu yang signifikan dan merupakan salah satu tantangan utama yang dihadapi oleh banyak perusahaan. Seperti yang dikemukakan oleh Emmett C. Murphy dan Mark A. Murphy, mendapatkan pelanggan baru jauh lebih sulit daripada menjaga pelanggan yang sudah ada [2], serta biaya yang dikeluarkan perusahaan lima kali lipat lebih banyak dibandingkan dengan memuaskan dan mempertahankan pelanggan lama [3].

Dalam mempertahankan customer, perusahaan telekomunikasi membutuhkan cara untuk memprediksi untuk mengetahui resiko kapan customer akan menjadi *churn*. Prediksi customer *churn* dapat dilakukan dengan teknik Machine Learning. Machine Learning merupakan yaitu metode ekstraksi data menjadi sebuah pola informasi tertentu [4].

Dan untuk melakukan pengenalan sebuah pola informasi dibutuhkan sebuah algoritma yang mampu untuk mengklasifikasikan customer *churn* atau tidak *churn*. Untuk itu penelitian ini menggunakan algoritma *K-nearest neighbors* (KNN). Algoritma KNN merupakan merupakan metode *algoritma*

supervised learning yang tujuannya untuk mendapatkan pola baru dalam sebuah data. Algoritma KNN termasuk salah satu pembelajaran mesin yang paling sederhana dimana sebuah objek diklasifikasikan berdasarkan kelas mayoritas sejumlah *K-nearest neighbors* [5].

Berdasarkan permasalahan tersebut pada penelitian ini akan membuat model *machine learning* menggunakan algoritma *K-nearest neighbors* yang diimplementasikan kepada sebuah website untuk memprediksi customer *churn* dan customer tidak *churn* guna untuk mempertahankan pelanggan pada perusahaan telekomunikasi.

2. TINJAUAN PUSTAKA

2.1. *Customer Churn*

Customer churn merujuk pada keadaan di mana pelanggan berhenti menggunakan layanan atau produk dari suatu perusahaan atau berpindah ke pesaing. Hal ini bisa disebabkan oleh berbagai faktor, seperti ketidakpuasan pelanggan, penawaran yang lebih baik dari pesaing, atau perubahan kebutuhan pelanggan. Memahami dan memprediksi customer churn menjadi penting bagi perusahaan untuk mengambil tindakan yang tepat guna mempertahankan pelanggan yang ada.

2.2. *K-Nearst Neighbor* (KNN)

metode K-Nearst Neighbor (KNN) merupakan salah satu metode klasifikasi terhadap sekumpulan data yang berdasarkan mayoritas dari kategori dan tujuannya untuk mengklasifikasikan obyek baru berdasarkan atribut dan sample sample dari training data. Sehingga target output yang diinginkan mendekati ketepatan dalam melakukan pengujian pembelajaran [6].

2.3. *Machine Learning*

Machine Learning adalah studi tentang algoritma untuk mempelajari bagaimana melakukan tugas-tugas tertentu yang dilakukan secara otomatis oleh orangorang. Di sini, belajar mengacu pada kemampuan untuk melakukan berbagai kegiatan yang sudah ada atau untuk mengekstrapolasi kesimpulan baru dengan benar dari berbagai pola yang diamati sebelumnya [7].

2.4. *Artificial Intelligence Project Cycle*

Artificial Intelligence (AI) Project Cycle adalah rangkaian tahapan yang digunakan dalam penerapan kecerdasan buatan untuk memecahkan masalah. Tahapan yang biasanya terdiri dari problem scoping, acquisition, data exploration, modelling, dan deployment. Problem scoping melibatkan identifikasi masalah yang akan diselesaikan dan tujuan yang ingin dicapai. Acquisition melibatkan pengumpulan data yang relevan untuk analisis. Data exploration melibatkan eksplorasi dan pemahaman data yang ada. Modelling melibatkan pengembangan model prediksi menggunakan algoritma machine learning seperti KNN. Deployment melibatkan implementasi model yang telah dikembangkan dan integrasinya ke dalam sistem yang sesuai.

2.5. Evaluasi dan Implementasi

Evaluasi model dilakukan untuk mengukur kinerja model prediksi churn yang telah dikembangkan. Evaluasi ini melibatkan pengujian model menggunakan data yang tidak digunakan dalam proses training dan validasi. Berdasarkan hasil evaluasi, dapat dihitung akurasi model, yaitu seberapa baik model dapat memprediksi churn dengan benar. Selain itu, juga dapat diperoleh informasi tentang jumlah prediksi churn yang benar dan yang salah, serta jumlah prediksi tidak churn yang benar dan yang salah. Implementasi model dilakukan dengan mengintegrasikannya ke dalam sebuah website yang menyediakan fitur-fitur seperti data churn dan tidak churn, input data pelanggan, data pelanggan churn, dan data pelanggan tidak churn. Website tersebut terhubung dengan database menggunakan server *local host*, sehingga memungkinkan perusahaan telekomunikasi untuk memprediksi customer *churn* dengan menggunakan model yang telah dikembangkan.

3. METODE PENELITIAN

Pada penelitian ini ada beberapa proses yang dilakukan diantaranya:

3.1. *Data Acquisition*

Proses mengumpulkan data-data yang dibutuhkan untuk membuat proyek AI. Hal ini merupakan dasar atau bahan yang selanjutnya diolah untuk dianalisis sesuai masalah dan diamati agar bisa menghasilkan solusi terbaik. Pengumpulan data yang dilakukan adalah dengan mengambil data sekunder yang didapatkan pada platform github dengan judul telco_dataset berformat csv. Dataset yang didapat berjumlah 6,950

3.2. *Data Exploration*

a. Missing Value
   Missing Value adalah hilangnya beberapa data yang telah diperoleh. Dalam dunia data science, missing value erat kaitannya dalam proses perselisihan data (data wrangling) sebelum nantinya akan dilakukan analisis dan prediksi data.
b. Outlier
   Outlier, yaitu pengamatan yang memiliki nilai relatif besar atau kecil dibandingkan dengan sebagian besar pengamatan, dapat berdampak besar atau mendominasi hasil percobaan. Teknik untuk untuk deteksi outlier mengetahui menggunakan Boxplot. Fitur yang terdapat outlier dapat dibersihkan menggunakan teknik z-score [8]

c. Label Encoding
Label encoding mengacu pada mengubah label menjadi bentuk numerik untuk mengubahnya menjadi bentuk yang dapat dibaca mesin. Algoritma pembelajaran mesin kemudian dapat memutuskan dengan cara yang lebih baik bagaimana label tersebut harus dioperasikan. Ini adalah langkah sesudah pemrosesan yang penting untuk kumpulan data terstruktur dalam pembelajaran yang diawasi.

d. Korelasi Heat Map
Hasil dari pembuatan program korelasi (menggunakan library seaborn) ditunjukan secara visual dan numerik dalam bentuk matrik heatmap, dengan tampilan warna menunjukan level dari nilai korelasi [9].

e. Memilih fitur
Pisahkan fitur dan label dalam dataset. Dimana pada fitur kolom gender sampai total charges dalam variabel X dan kolom churn label dalam variable Y.

f. Balancing data
Balancing merupakan merupakan metode untuk mengatasi data yang tidak seimbang dengan cara meningkatkan jumlah sampel kelas minoritas sehingga setara dengan kelas mayoritas menggunakan fungsi Smotenc [10].

g. Standarisasi data
Standarisasi data merupakan mentransformasi atribut data ke distribusi normal (normally distributed Gaussian). Dimana nilai rata-rata nya 0 dan standar deviasi nya 1. Standarisasi data ini bermanfaat untuk algoritma Machine Learning dimana data diasumsikan memiliki distribusi normal. Sehingga data yang sudah distandarisasi, harapannya untuk algoritma Machine Learning tersebut menghasilkan performansi yang tinggi. Library yang digunakan pada standarisasi data ini menggunakan scikit learn preprocessing, yang dipakai adalah *Standard Scaler* [11].

h. Split train set dan test set
Split data adalah membagi data menjadi dua bagian dimana train set dataset yang kita latih untuk membuat prediksi atau menjalankan fungsi dari sebuah algoritma Machine Learning dan test set bagian dataset yang akan dites untuk melihat keakuratannya, atau melihat performanya [12].

3.3. *Modelling*

K-Nearst Neighbor (KNN) merupakan algoritma untuk metode klasifikasi terhadap suatu data berdasarkan pembelajaran data yang sudah terklasifikasikan atau supervised learning.

Stratified kfold adalah teknik untuk cross-validation yang bertujuan untuk memperoleh hasil akurasi yang maksimal [13].

GridSearchCV, Grid search merupakan standar pemilihan kombinasi model dan hyperparameter [14].

3.4. Evaluasi

Pada fase ini dilakukan evaluasi atas model-model yang telah dibuat. Evaluasi dilakukan untuk memastikan bahwa ada model yang telah memenuhi yang telah ditentukan diawal. Pada akhir evaluasi ini, membandingkan dua model yang telah dilakukan yaitu KNN dan SVM.

3.5. *Deployment*

Pada fase ini dilakukan penerapan model terhadap media yang dapat memungkinkan pengguna memahami cara menggunakan model yang telah dibuat. Bentuk kegiatan deployment didasarkan kembali pada daftar kebutuhan yang telah ditentukan pada fase awal. Bentuk kegiatan deployment bisa dilakukan dalam berbagai bentuk mulai dari sebatas laporan, hingga penerapan sistem untuk interaksi langsung antara sistem dan pengguna. Deployment yang dilakukan diimplementasikan ke sebuah website menggunakan framework *flask* dan *bootstrap* yang terhubung dengan database server localhost.

4. HASIL DAN PEMBAHASAN

4.1. *Data Acquisition*

Pengumpulan data yang dilakukan adalah dengan mengambil data sekunder yang didapatkan pada platform github dengan judul *telco_dataset* berformat csv. Dataset yang didapat berjumlah 6,950 data dengan kolom *UpdateAt, CustomerID, Gender, SeniorCitizen, Partner, Tenure, PhoneService, StreamingTV, Internet Service, PapersBilling, MonthlyCharges, TotalCharges*, dan *Churn*.

	UpdatedAt	customerID	gender	SeniorCitizen	Partner	tenure	PhoneService	StreamingTV	InternetService	PaperlessBilling	MonthlyCharges	TotalCharges	Churn
2	202006	45759018157	Female	No	Yes	1	No	No	Yes	Yes	29.85	29.85	No
3	202006	45315483266	Male	No	Yes	60	Yes	No	No	Yes	20.5	1198.8	No
4	202006	45236961615	Male	No	No	5	Yes	Yes	Yes	No	104.1	541.9	Yes
5	202006	45929827382	Female	No	Yes	72	Yes	Yes	Yes	Yes	115.5	8312.75	No
6	202006	45305082233	Female	No	Yes	56	Yes	Yes	Yes	No	81.25	4620.4	No
7	202006	45072364214	Male	No	No	44	Yes	Yes	Yes	Yes	85.25	3704.15	No
8	202006	45410681487	Male	No	Yes	39	Yes	Yes	Yes	No	80	3182.95	Yes

Gambar 1. Dataset customer churn

Dataset yang sudah dikumpulkan tersebut kemudian akan simpan pada google drive, nantinya akan masuk ke data *Exploration*.

4.2. *Data Exploration*

Load Test, Dataset file *telco_dataset.csv* yang sudah ditempatkan pada google drive kemudian akan di load, kemudian akan ditempatkan pada suatu variabel yaitu *datacustel*. Berikut ini adalah kode program yang digunakan.

```
#Membaca Dataset
datacustel = pd.read_csv('/content/drive/MyDrive/PROJEK AKHIR/APP/DATASET/telco_dataset.csv')
datacustel.head()
```

Gambar 2. Kode program load test

Memeriksa missing value, Dataset tersebut kemudian akan diperiksa untuk memastikan dataset dalam keadaan lengkap atau tidak ada nilai yang hilang. Berikut ini adalah kode program yang digunakan.

```
[ ] # Mengecek Mising value
    datacustel.isnull().sum()

    UpdatedAt          0
    customerID         0
    gender             0
    SeniorCitizen      0
    Partner            0
    tenure             0
    PhoneService       0
    StreamingTV        0
    InternetService    0
    PaperlessBilling   0
    MonthlyCharges     0
    TotalCharges       0
    Churn              0
    dtype: int64
```

Gambar 3. Kode program memeriksa missing value

Berdasarkan kode program pada gambar 3, tidak ada nilai yang hilang pada setiap kolom nya.

Mengecek *outlier* sekaligus menangani *outlier*. Dataset tersebut kemudian akan diperiksa untuk memastikan dataset dalam keadaan terbebas dari *outlier*. Berikut ini adalah kode program yang digunakan.

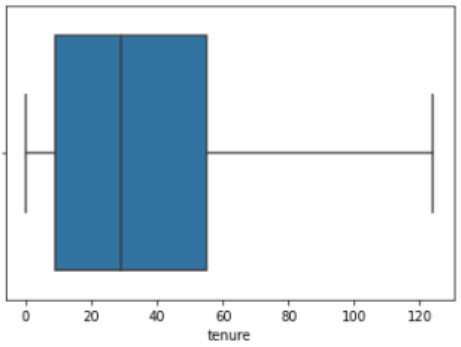
```
# Menampilkan statistik dari dataset
datacustel.describe(include='all')
```

SeniorCitizen	Partner	tenure	PhoneService	StreamingTV	InternetService	PaperlessBilling	MonthlyCharges	TotalCharges	Churn
6950	6950	6950.000000	6950	6950	6950	6950	6950.000000	6950.000000	6950
2	2	NaN	2	2	2	2	NaN	NaN	2
No	No	NaN	Yes	No	Yes	Yes	NaN	NaN	No
5822	3591	NaN	6281	4279	5445	4114	NaN	NaN	5114
NaN	NaN	32.423165	NaN	NaN	NaN	NaN	64.992201	2286.058750	NaN
NaN	NaN	24.581073	NaN	NaN	NaN	NaN	30.032040	2265.702553	NaN
NaN	NaN	0.000000	NaN	NaN	NaN	NaN	0.000000	19.000000	NaN
NaN	NaN	9.000000	NaN	NaN	NaN	NaN	36.462500	406.975000	NaN
NaN	NaN	29.000000	NaN	NaN	NaN	NaN	70.450000	1400.850000	NaN
NaN	NaN	55.000000	NaN	NaN	NaN	NaN	89.850000	3799.837500	NaN
NaN	NaN	124.000000	NaN	NaN	NaN	NaN	169.931250	8889.131250	NaN

Gambar 4. Kode program mengecek outlier berbasis deskripsi statistik

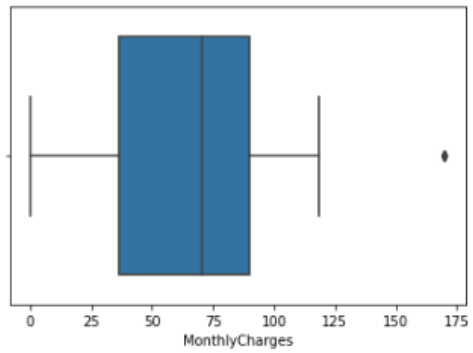
Berdasarkan kode program pada gambar 4, Kolom yang terdapat nilai NaN adalah kolom bertipe kategorikal sehingga yang akan selanjutnya diperiksa untuk memastikan terbebas dari *outlier* kolom yang memiliki nilai min dan max yang terlalu jauh perbandingan nya, sehingga kolom *tenure*, *monthlycharges*, dan *totalcharges* selanjutnya akan diperiksa menggunakan boxplot untuk memastikan terbebas dari *outlier*. Berikut ini adalah kode program yang digunakan.

```
[ ] # memeriksa outlier pada kolom tenure
    sns.boxplot(datacustel['tenure'])
    plt.show()
```



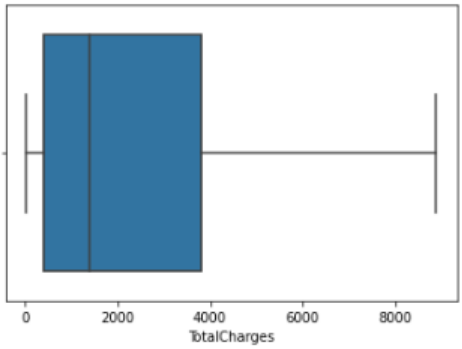
Gambar 5. Kode program mengecek outlier boxplot kolom tenure

```
[ ] # memeriksa outlier pada kolom MonthlyCharges
    sns.boxplot(datacustel['MonthlyCharges'])
    plt.show()
```



Gambar 6. Kode program mengecek outlier boxplot kolom monthlycharges

```
[ ] # memeriksa outlier pada kolom TotalCharges
    sns.boxplot(datacustel['TotalCharges'])
    plt.show()
```



Gambar 7. Kode program mengecek outlier boxplot kolom totalcharges

Berdasarkan kode program pada gambar 6. kolom *MonthlyCharges* terdapat *outlier*, sehingga diperlukannya pembersihan *outlier* tersebut dengan menggunakan Z-score kurang dari 3. Data yang

memiliki nilai Z-score lebih dari 3 tidak akan dimasukan kedalam data yang selanjutnya akan diproses pada tahap berikutnya. Berikut ini adalah kode program yang digunakan.

```
[ ] #kolom yang terdapat outlier
    kolom_numerik = ['MonthlyCharges']
    #pemebersihan outlier
    datacustel = datacustel[(np.abs(stats.zscore(datacustel[kolom_numerik])) < 3).all(axis=1)]
```

Gambar 8. Kode program pembersihan outlier kolom monthlycharges

Melakukan label encoding, Dataset tersebut kemudian akan dilakukanya label encoding. Label encoding dibutuhkan karena pada Model AI (*Machine Learning* dan *Deep Learning*) hanya menerima nilai numerik sebagai input, sehingga kolom yang terdapat kolom kategorik seperti *gender* yang awal nya male menjadi 0 dan *female* menjadi 1, kemudian pada kolom kategorik *SeniorCitizen, Partner, StreamingTV, InternetService, PapersBilling* dan *Churn* akan diubah yang awal nya *No* menjadi 0 dan *Yes* menjadi 1. Berikut ini adalah kode program yang digunakan.

```
[ ] # label encoding secara manual dengan memanfaatkan fuction map()

    # kolom gender
    datacustel['gender'] = datacustel['gender'].map({'Male':0, 'Female':1})

    # kolom SeniorCitizen
    datacustel['SeniorCitizen'] = datacustel['SeniorCitizen'].map({'No':0, 'Yes':1})

    # kolom Partner
    datacustel['Partner'] = datacustel['Partner'].map({'No':0, 'Yes':1})

    # kolom PhoneService
    datacustel['PhoneService'] = datacustel['PhoneService'].map({'No':0, 'Yes':1})

    # kolom StreamingTV
    datacustel['StreamingTV'] = datacustel['StreamingTV'].map({'No':0, 'Yes':1})

    # kolom InternetService
    datacustel['InternetService'] = datacustel['InternetService'].map({'No':0, 'Yes':1})

    # kolom PaperlessBilling
    datacustel['PaperlessBilling'] = datacustel['PaperlessBilling'].map({'No':0, 'Yes':1})

    # kolomfitur Churn
    datacustel['Churn'] = datacustel['Churn'].map({'No':0, 'Yes':1})
```

Gambar 9. Kode program label encoding

Menampilkan korelasi atau hubungan pada setiap kolom, Kemudian mengetahui keeratan hubungan antara beberapa kolom pada dataset tersebut. Lalu memilih kolom/fitur yang akan di training, kemudian memilih kolom yang penting yaitu pada kolom *gender, pertner, seniorcitizen, tenure, phoneservice, streamingtv, internetservice, papersbiling, monthlycharges* dan *totalcharges.* Mengecek data tidak seimbang/imbalancing data pada kolom *churn*(Y), memeriksa ketidakseimbangan data pada kolom *churn*, ketidakseimbangan data dapat menyebabkan bias. Bias adalah kecenderungan model yang memberikan hasil prediksi label mayoritas. Berikut ini adalah kode program yang digunakan.

```
[19] #mengecek data fitur churn(Y)
     np.unique(Y, return_counts=True)

     (array([0, 1]), array([5111, 1836]))
```

Gambar 10. Kode program memeriksa ketidakseimbangan data kolom churn.

Berdasarkan kode program pada gambar 10. terdapat ketidakseimbangan data diamana class 0 memiliki nilai berjumlah 5111 sedangkan class 1 berjumlah 1836. Sehingga berdasarkan permasalahan tersebut diperlukan nya penyeimbangan data resampling menggunakan fungsi *Smotenc*. Berikut kode program yang digunakan.

```
[20] #import SMOTENC
     from imblearn.over_sampling import SMOTENC

[21] #melakukan imbalancing data dengan Smotenc
     sm = SMOTENC(random_state=42, categorical_features=[2,3,4,6,7,8,9])
     print(sm)
     X_res, y_res = sm.fit_resample(X, Y)

     SMOTENC(categorical_features=[2, 3, 4, 6, 7, 8, 9], random_state=42)
```

Gambar 11. Menangani Ketidakseimbangan Data

```
[22] #setelah dilakukan nya balancing data penyeimbangan data
     np.unique(y_res, return_counts=True)

     (array([0, 1]), array([5111, 5111]))
```

Gambar 12. Setelah Penyeimbangan Data

Standarisasi data, Standarisasi data diperlukan untuk menyeragamkan nilai-nilai pada data yang memungkinkan pemrosesan data yang efisien. Berikut kode program yang digunakan.

```
#import standarscaler
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
scaler = scaler.fit(X_res)
X = scaler.transform(X_res)
print(X)

[[ 1.01379104 -0.49519558  1.11783713 ...  0.74333817 -1.34324427
  -0.8891082 ]
 [-0.98639657 -0.49519558  1.11783713 ...  0.74333817 -1.67075589
  -0.36151424]
 [-0.98639657 -0.49519558 -0.8945847  ... -1.34528272  1.25758334
  -0.65799952]
 ...
 [ 1.01379104 -0.49519558 -0.8945847  ...  0.74333817  0.01584141
  -0.75740717]
 [-0.98639657 -0.49519558 -0.8945847  ...  0.74333817  0.19798691
  -0.87024219]
 [ 1.01379104 -0.49519558 -0.8945847  ...  0.74333817 -0.65844723
  -0.82752285]]
```

Gambar 13. Standarisasi data

Split train set dan test set, membagi dataset menjadi dua bagian yakni bagian yang digunakan untuk training data dan untuk testing data dengan proporsi 80% kedalam train dan 20% ke dalam test.

```
[24] #import slpit data
     from sklearn.model_selection import train_test_split

     #inisialisai data X_train, X_test, y_train, y_test, data dibagi ke dalam 80% train, 20% test
     X_train, X_test, y_train, y_test = train_test_split(X, y_res, test_size=0.2, random_state=42)

     print('Train set size : ', X_train.shape, y_train.shape)
     print('Test set size  : ', X_test.shape, y_test.shape)

     Train set size :  (8177, 10) (8177,)
     Test set size  :  (2045, 10) (2045,)
```

Gambar 14. Split data

4.3. Modelling

Dalam membangun proyek berbasis AI, diperlukan untuk bekerja dengan algoritma AI. Dalam memilih algoritma, perlu melihat pendekatan model apakah *rule-based* atau pendekatan learning. Jika telah

menemukan model AI yang tepat, maka akan dilakukan pelatihan model dengan data yang sudah terkumpul sebelumnya. Adapun tuning *hyperparameters* program pada algoritma ini yang dapat ditampilkan pada kode program berikut:

```
[27] #Parameter
     hyperparameters_knn = {'n_neighbors': list(range(2, 21)),
                            }
     cv = StratifiedKFold(n_splits=5, random_state=0, shuffle=True)
```

Gambar 15. Hyperparameter model KNN

Berdasarkan kode program pada gambar 15. dimana pada tuning hyperparameters terdapat *n_neighbors* yang digunakan untuk mencari nilai dari K dengan nilai range dari 2 sampai 21. Kemudian *StratifiedKFold* dengan parameter *n_splits* = 5, *random_state*= 0, *shuffle=True*, digunakan untuk mengacak data, setelah itu memecah data menjadi n_splits. Kemudian parameter tersebut akan dicari yang terbaik menggunakan fungsi *GridSearchCV*.

```
[ ] model_knn = GridSearchCV(KNeighborsClassifier(), hyperparameters_knn, cv=cv, n_jobs=-1, verbose=10)

    model_knn.fit(X_train, y_train)
    print('Setting model KNN terbaik :', model_knn.best_params_)

    Fitting 5 folds for each of 19 candidates, totalling 95 fits
    Setting model KNN terbaik : {'n_neighbors': 7}
```

Gambar 16. GridSearch model KNN

Berdasarkan kode program gambar 16 dimana grid search dengan parameter nya yaitu *KNeighborsClassifier*() digunakan sebagai model nya, hyperparameters_knn dengan tuning hyperparameters, cv=cv menggunakan fungsi *StratifiedKFold*, *n_jobs*=-1, *verbose*=10, menghasilkan parameter terbaik pada parameter *n_ neighbors* 7.

4.4. Evaluation

Evaluasi model dilakukan dengan melakukan prediksi terhadap data testing menggunakan model KNN yang terbentuk. Evaluasi yang dilakukan menggunakan *classification report* dan mengetahui test set *accuracy* dan train set *accuracy* untuk menentukan apakah model *overfitting, goodfit, underfitting.* Berikut kode program nya.

```
y_pred = model_knn.predict(X_test)
print(classification_report(y_test, y_pred))

              precision    recall  f1-score   support

           0       0.82      0.81      0.81      1045
           1       0.80      0.82      0.81      1000

    accuracy                           0.81      2045
   macro avg       0.81      0.81      0.81      2045
weighted avg       0.81      0.81      0.81      2045

[ ] print("Train set Accuracy : ", metrics.accuracy_score(y_train, model_knn.predict(X_train)))
    print("Test set Accuracy : ", metrics.accuracy_score(y_test, y_pred))

    Train set Accuracy :  0.8404060168766051
    Test set Accuracy :  0.8112469437652812
```

Gambar 17. Kode program evaluasi model

Berdasarkan kode program gambar 17 model yang dihasilkan mendapatkan *accuracy* 81% dengan, *precision* 81%, *recall* di 81%, dan f1 score 81%. Kemudian model tersebut bisa dikatakan godfit karena pada train set *accuracy* mendapatkan *accuracy* 84% dan test set *accuracy* 81% dengan demikian nilai kedua nya hanya berjarak 3%. Kemudian berdasarkan hasil confusion matrix untuk prediksi customer *churn*, model memprediksi pelanggan *churn* yang benar adalah 817 salah 183 prediksi, lalu memprediksi pelanggan tidak *churn* yang benar adalah 842 dan salah 203.

```
#Menampilkan Confuison Matrix
confusion_matrix(y_test, y_pred)

array([[842, 203],
       [183, 817]])
```

Gambar 18. Confusion matrix

Setelah menentukan model dengan parameter terbaik selanjutnya menyimpan model dan *scaler* dengan format *pkl* yang nantinya akan diintegrasikan ke tahap deployment.

4.5. Deployment

Proses deployment yang dilakukan mengimplementasikan nya ke suatu website dengan *framework flask* sebagai *beckend* dan *bootstrap* sebagai frontend yang terintegrasi dengan database mysql serta beberapa library tambahan seperti pickle dan sklearn. Adapun tahap-tahap yang dilakukan yaitu:

4.6. Pembuatan template website

Pembuatan template menggunakan framework *bootstrap* V5. Template tersebut meliputi beberapa halaman seperti halaman index.html sebagai halaman utama, halaman pelanggan.html sebagai halaman untuk memprediksi, halaman pelanggan_aktif.html dan pelanggan_nonaktif.html.

4.7. Pembuatan database

Database yang digunakan menggunakan Mysql phpmyadmin dengan nama database telekom dan nama table pelanggan. Table tersebut memiliki kolom atau nilai seperti *UpdatedAt, CustomerID, Name, Gender, SeniorCitizen, Partner, Tenure, PhoneService, StreamingTV, InternetService, PaperlessBilling, MonthlyCharges, TotalCharges, Churn.*

4.8. Mengintegrasikan database dengan website

Mengintegrasikan database pada *flask* menggunakan *library flask_mysqldb* dimana *Host* nya yaitu *localhost*, user nya *root* dan *password* dikosongkan dan nama databasenya adalah telekom.

```
#connect database
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'telekom'
mysql = MySQL(app)
```

Gambar 19. Kode program koneksi ke database

4.9. Mengintegrasikan model dengan website

Mengintegrasikan model dengan website dengan membuat fungsi seperti pada file *model.py* terdapat fungsi *load*() untuk *load* model, kemudian *prediksi_data()* untuk memprediksi data yang dimasukan. Lalu pada *app.py* terdapat fungsi predict untuk mengambil dan merubah data kategorik menjadi numerik dan memanggil fungsi dari *prediksi_data ()*. Nantinya data tersebut akan dimasukan kedalam database kolom *churn*.

```
import pickle

# global variable
global model, scaler


#load model dan scaler
def load():
    global model, scaler
    model = pickle.load(open('model/model_knn.pkl', 'rb'))
    scaler = pickle.load(open('model/scaler_churn.pkl', 'rb'))

#fungsi prediksi data dan mereturn hasil
def prediksi_data(data):
    data = scaler.transform(data)
    prediksi = int(model.predict(data))
    if prediksi == 0:
        hasil_prediksi = "No"
    else:
        hasil_prediksi = "Yes"
    return hasil_prediksi
```

Gambar 20. Kode program prediksi data

```
@app.route("/predict", methods=["POST"])
def predict():
    cursor = mysql.connection.cursor()
    cursor.execute('''SELECT * FROM pelanggan ''')
    pelanggan_predik = cursor.fetchall()

    #merubah data kategorik menjadi numerik
    for f in pelanggan_predik:
        if (f[3] == 'Female'):
            gender_pred = 0
        else:
```

Gambar 21. Kode program data kategorik menjadi numerik

```
        data = [[gender_pred, senior_pred, partner_pred, tenure_pred, phoneservice_pred,
        stream_pred, internet_pred, papersbil_pred, papersbil_pred, monthnly_pred]]

        #prediksi data
        prediction_result = prediksi_data(data)

        #memasukan data Yes atau No kedalam kolom churn sesuai dengan ID customer
        cursor.execute('''UPDATE pelanggan SET Churn = %s WHERE CustomerID = %s''',
                       (prediction_result, CustomerID))
        mysql.connection.commit()
    return redirect(url_for('pelanggan'))
```

Gambar 22. Kode program menyimpan data ke database

4.10. Deskripsi Website

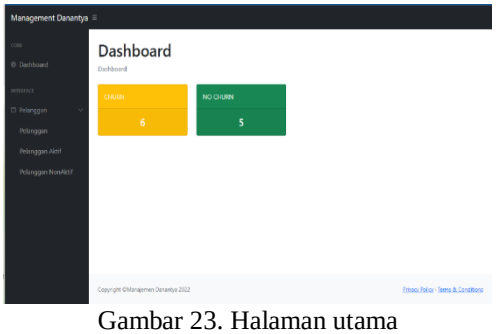
Management Danantya merupakan sebuah aplikasi berupa web secara online yang digunakan untuk memprediksi customer *churn* dengan tambahan fitur layanan yang dikembangkan dengan Teknologi Artificial Intelligence (AI) khususnya pada bidang Data *Science*. Website ini dapat digunakan pada perusahaan telekomunikasi untuk memprediksi customer tersebut apakah *churn* atau tidak *churn*. Hasil dari prediksi tersebut akan menjadi suatu pertimbangan bagi perusahaan untuk membuat suatu kebijakan apa yang akan diambil. Tujuan utama dalam pembuatan aplikasi ini yaitu untuk mengetahui pelanggan pada suatu perusahaan itu loyal atau *churn* dengan indikator tertentu. Berikut adalah indikatornya antara lain:

1. *SeniorCitizen*: Apakah usia customer lebih dari 60 tahun (Yes, No)
2. *Partner*: *Apakah* customer memiliki pasangan (Yes, No)
3. *PhoneService*: Apakah *customer menggunakan* layanan telepon (Yes, No)
4. *InternetService*: *Apakah customer* menggunakan layanan internet service provider (Yes, No)
5. *StreamingTV*: Apakah *customer* menggunakan layanan *StreamingTV* (Yes, No)
6. *PaperlessBilling*: *Apakah customer* membayar dengan eBilling/ePayment (Yes, No)
7. *Tenure*: Lama *customer berlangganan* (Bulan)
8. *MonthlyCharges*: Total biaya perbulan yang dikeluarkan customer
9. *TotalCharges*: Total biaya perbulan yang dikeluarkan berdasarkan tenure.

Churn: Apakah customer *Churn* (Yes, No). Di dalam fitur ini cukup untuk memasukkan kategori data yang ada pada kolom mulai dari SeniorCitizen hingga ke TotalCharges kemudian dilakukanlah sebuah prediksi berdasarkan indikator-indikator tersebut sehingga terciptalah sebuah hasil.

4.11. Halaman utama

Halaman utama merupakan tampilan awal atau pertama pada saat membuka website management danantya.

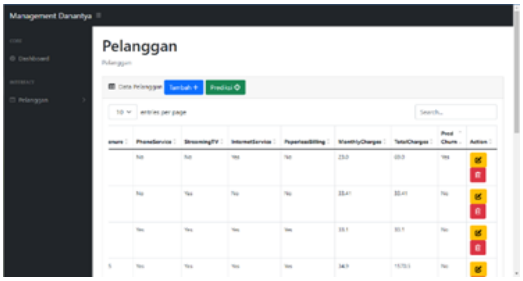


Gambar 23. Halaman utama

4.12. Halaman pelanggan

Pada menu halaman pelanggan akan menampilkan section pelanggan yang meliputi pelanggan (secara keseluruhan), pelanggan aktif, dan pelanggan non aktif. Pada halaman ini terdapat fungsi yang dapat digunakan untuk mencari data, mengedit

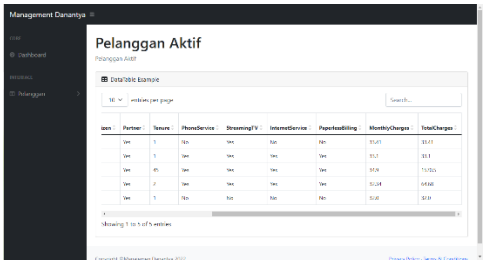
data, menghapus data, dan juga dapat melakukan pengurutan data. Pada section pelanggan itu berisi data pelanggan secara keseluruhan beserta dengan prediksinya.



Gambar 24. Halaman pelanggan

4.13. Halaman pelanggan aktif

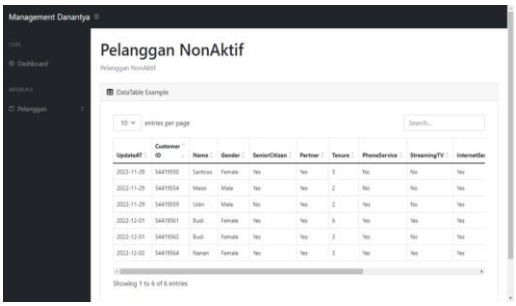
Halaman pelanggan aktif ini merupakan tampilan dimana data-data pelanggan yang masih aktif.



Gambar 25. Halaman pelanggan aktif

4.14. Halaman pelanggan non aktif

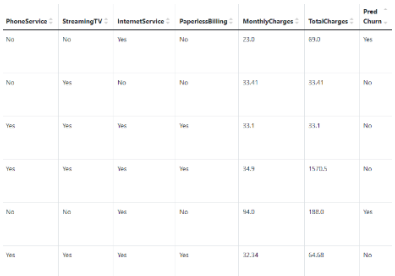
Halaman pelanggan Nonaktif ini merupakan tampilan dimana data-data pelanggan yang sudah tidak aktif.



Gambar 26. Halaman pelanggan non aktif

4.15. Performa Aplikasi

Adapun beberapa hasil pengujian model AI sebagai berikut:



PhoneService	StreamingTV	InternetService	PaperlessBilling	MonthlyCharges	TotalCharges	Pred Churn
No	No	Yes	No	23.0	89.0	Yes
No	Yes	No	No	33.41	33.41	No
Yes	Yes	Yes	Yes	53.1	53.1	No
Yes	Yes	Yes	Yes	54.9	1570.5	No
No	No	Yes	No	94.0	188.0	Yes
Yes	Yes	Yes	Yes	32.34	64.68	No

Gambar 27. Data pengujian hasil prediksi

	precision	recall	f1-score	support
0	0.82	0.81	0.81	1045
1	0.80	0.82	0.81	1000
accuracy			0.81	2045
macro avg	0.81	0.81	0.81	2045
weighted avg	0.81	0.81	0.81	2045

Gambar 28. Classification report

Berdasarkan hasil evaluasi tersebut model dari KNN ini memiliki akurasi 81% saat memprediksi data yang akan dimasukan.

4.16. Kelebihan dan Kekurangan

1. Kelebihan
   Kelebihan pada website ini yaitu:
   a. Mampu memprediksi banyak data customer sekaligus
   b. Data customer mampu diedit kembali sekaligus dapat memprediksi jika data tersebut sudah edit dengan data yang terbaru
   c. Data tersebut data dihapus
   d. Terdapat informasi mengenai berapa jumlah data customer yang churn atau tidak churn pada halaman utama
   e. Terdapat sebuah halaman yang sudah menyeleksi pelanggan yang churn dan tidak churn
   f. Dapat mencari data tersebut melalui fitur pencarian
2. Kelemahan
   Kelebihan pada website ini yaitu:
   a. Memiliki fitur-fitur website yang sedikit
   b. Akurasi model AI hanya 81%.
   c. Data yang digunakan masih data perusahaan asing sehingga masih menggunakan bahasa inggris

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan, tuning hyperparameter pada algoritma KNN menggunakan parameter *n_neighboard* dan fungsi *StratifiedKFold* dengan menggunakan dataset proyek ini, mendapatkan setting parameter *n_neighboard* terbaik pada nilai 7.

Hasil dari evaluasi yang telah dilakukan pada modelling menggunakan algoritma KNN untuk melakukan prediksi customer *churn* mendapatkan *accuracy* 81%, precision 81%, *recall*l 81%, dan f1-score 81% dan model tersebut memiliki dapat dikatakan goodfit berdasarkan hasil bukti train set *Accuracy*: 84% dan Test set *Accuracy*: 81% dengan perbandingan kedua nilai tersebut tidak terlalu jauh hanya berbeda 0.3. Kemudian berdasarkan hasil confusion matrix untuk prediksi customer *churn*, model memprediksi pelanggan *churn* yang benar adalah 817 prediksi dan yang salah memiliki jumlah sebanyak 183, dan pelanggan tidak *churn* yang benar adalah 842 prediksi dan yang salah memiliki jumlah sebanyak 203.

Deployment yang telah dilakukan diimplementasikan sebuah website dengan framework *flask* dan bootstrap. Website tersebut mampu memprediksi banyak data customer sekaligus, data customer mampu diedit kembali sekaligus dapat memprediksi jika data tersebut sudah edit dengan data yang terbaru, data tersebut dapat dihapus, terdapat informasi mengenai berrapa jumlah data customer yang *churn* atau tidak *churn* pada halaman utama.

Saran yang penulis sampaikan, Modelling menggunakan algoritma pada proyek ini dapat ditingkatkan kembali pada *accuracy*, precision, *recall*l, dan f1-score seperti mengubah tuning hyper parameter pada model atau mengganti algoritma nya, tampilan dan fitur-fitur pada website yang dapat dikembangkan menjadi lebih banyak dan informatif, seperti menambahkan fitur grafik pelanggan yang *churn* dari waktu ke waktu, menambahkan fitur rekomendasi promo yang dapat diberikan kepada pelanggan yang terprediksi *churn*.

DAFTAR PUSTAKA

[1] M. Herawati, I. Mukhlash, & I. L. Wibowo. (2016). Prediksi Customer Churn Menggunakan Algoritma Fuzzy Iterative Dichotomiser 3. Mathematics and Its Applications, 13(1), 23-36.

[2] E. C. Murphy, & M. A. Murphy. (2002). Leading on the Edge of Chaos: The 10 Critical Elements for Success in Volatile Times. USA: Prentice Hall Press.

[3] H. K. Pambudi, P. G. A. Kusuma, F. Yulianti, & K. A. Julian. (2020). Prediksi Status Pengiriman Barang Menggunakan Metode Machine Learning. Ilmiah Teknologi Informasi Terapan, 6(2), 2686-0333.

[4] Rachmi, A. N. (2020). Implementasi Metode Random Forest Dan Xgboost Pada Klasifikasi Customer Churn. Yogyakarta.

[5] Sabilla, W. I., & Putri, T. E. (2017). Prediksi Ketepatan Waktu Lulus Mahasiswa Dengan K-Nearest Neighbordan Naïve Bayes Classifier [Predicting the Timeliness of Students' Graduation Using K-Nearest Neighbors and Naïve Bayes Classifier]. Komputer Terapan, 3(2), 233-240.

[6] Putra, P., Pardede, A. M. H., & Syahputra, S. (2022). Analisis Metode K-Nearest Neighbour (KNN) dalam Klasifikasi Data Iris Bunga [Analysis of K-Nearest Neighbour (KNN) Method in Iris Flower Data Classification]. Jurnal Teknik Informatika Kaputama (JTIK), 6(1), 1-2022.

[7] Shalev-Shwartz, S., & Ben-David, S. (2013). Understanding machine learning: From theory to algorithms (Vol. 9781107057). Cambridge University Press. DOI: 10.1017/CBO9781107298019.

[8] Whendasmoro, R. G., & Joseph. (2022). Analisis penerapan normalisasi data dengan menggunakan Z-Score pada. Jurnal Riset Komputer, 9(4), 872-876.

[9] Hastomo, W., Karno, A. S. B., Kalbuana, N., Nisfiani, E., & Lussiana. (2021). Optimasi Deep Learning Untuk Prediksi Saham Di Masa Pandemi Covid-19. Edukasi Dan Penelitian Informatika, 7(2).

[10] Wijaya, J., Soleh, A. M., & Rizki, A. (2018). Penanganan Data Tidak Seimbang pada Pemodelan Rotation Forest untuk Keberhasilan Studi Mahasiswa Program Magister IPB. Statistika, 2(2), 32-40.

[11] Widodo, R. B., Swastika, W., Setiawan, H., & Subianto, M. (2021). Studi Pemrosesan Data Pengenalan Gestur Tangan Menggunakan Metode KNN. In Conference on Innovation and Application of Science and Technology (pp. 277-280).

[12] Herlambang, M. (2018). Training Dan Test Set. Epam International. Retrieved 5 Desember 2022 from https://www.megabagus.id/training-set-test-set

[13] Khalimi, A. M. (2020). Pengujian Data Dengan Cross Validation. April 2020. Retrieved 5 Desember 2022 from https://www.pengalaman-edukasi.com/2020/04/apa-itu-k-fold-cross-validation.html

[14] Michael, A. (2022). Komparasi Kombinasi Pre-Trained Model Dengan SVM Pada Klasifikasi Kematangan Kopi Berbasis Citra. Dynamic Saint, 7(1), 42-47.