Classification of Vacational High School Graduates' Ability in Industry using Extreme Gradient Boosting (XGBoost), Random Forest And Logistic Regression
[Klasifikasi Kemampuan Lulusan SMK di Industri Menggunakan Extreme Gradient Boosting (XGBoost), Random Forest dan Logistic Regresssion]

Afikah Agustiningsih[1)], Yulian Findawati[2)] , Irwan Alnarus Kautsar[3)]

[1,2,3)]Program Studi Teknik Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia
Email Penulis Korespondensi: 191080200101@umsida.ac.id

Abstract. *The world of education is one of the main sources in presenting Human Resources. Vocational High School (SMK) is one of the school levels that presents various majors that are ready to compete in the industrial world. therefore a school institution needs to have a system to find out how much the quality of education provided to students is able to compete in the industrial world. The goal is that school institutions can strategize to produce better quality students in the following year. In this study there are 4 classes, namely working, not working, students, and entrepreneurs. There are several stages in building a classification system including the preprocessing, processing and evaluation stages. This research uses three machine learning algorithms namely XGBoost, Random Forest, and Logistic Regression. The results of the three methods get an accuracy score of 67% produced by the XGBoost and Random Forest algorithms, 50% accuracy score by Logistic Regression.*

Keywords – *Classification; graduate quality; machine learning; vocational high school (SMK).*

Abstrak. *Dunia pendidikan merupakan salah satu sumber utama dalam menghadirkan Sumber Daya Manusia. Sekolah Menengah Kejuruan (SMK) merupakan salah satu tingkatan sekolah yang menghadirkan berbagai jurusan yang siap untuk bersaing di dunia industri. oleh karena itu suatu lembaga sekolah perlu mempunyai sistem untuk mengetahui seberapa besar kualitas pendidikan yang diberikan ke peserta didik agar mampu bersaing di dunia industri. Tujuannya adalah agar lembaga sekolah dapat menyusun strategi agar menghasilkan kualitas peserta didik yang lebih baik di tahun berikutnya. Pada penelitian ini terdapat 4 kelas yaitu bekerja, belum bekerja, mahasiswa, dan wirausaha. Terdapat beberapa tahapan dalam membangun sistem klasifikasi diantaranya yaitu tahap prepocessing, processing dan evaluasi. Penelitian ini menggunakan tiga algoritma machine learning yaitu XGBoost, Random Forest, dan Logistic Regression. Hasil dari ketiga metode tersebut mendapatkan skor akurasi 67% yang dihasilkan oleh algoritma XGBoost dan Random Forest, skor akurasi 50% Oleh Logistic Regression.*

Kata Kunci – *Klasifikasi; Kualitas lulusan; machine learning; SMK*

I. PENDAHULUAN

Sumber Daya Manusia (SDM) merupakan aset yang paling penting dalam kemajuan pembangunan suatu Negara, khususnya pada suatu organisasi/lembaga pendidikan, sehingga sangat penting agar sumber daya manusia yang ada dapat direncanakan sebaik-baiknya, demi terwujudnya tujuan organisasi atau lembaga. Sumber Daya Manusia disadari sepenuhnya memiliki dampak yang sangat besar dan dianggap sebagai kunci utama dalam meningkatkan mutu pendidikan [1]. Salah satu penyebab baik tidaknya SDM suatu Negara adalah dengan melihat kualitas pendidikan.

Pendidikan memegang peranan penting dalam kemajuan suatu Negara, dimana hal ini menjadi suatu kepentingan bagi negara yang ining berkembang, maju serta mampu bersaing secara global [2].

Pemerintah mempersiapkan Sekolah Menengah Kejuruan (SMK) untuk mengatasi permasalahan pengangguran yang ada dan meningkatkan taraf hidup serta mendorong masyarakat untuk memiliki harapan karir yang lebih menonjol [3]. Oleh karena itu diperlukan adanya sistem yang dapat membantu pihak sekolah untuk melakukan klasifikasi lulusan peserta didik sebagai bahan untuk evaluasi lulusan di tahun berikutnya.

Beberapa penelitian terdahulu yang mengkaji mengenai kemampuan lulusan SMK pernah dilakukan sebelumnya. Diantaranya adalah penelitian yang dilakukan oleh Lianny Wydiastuty Kusuma dengan judul "Prediksi kemampuan lulusan SMK untuk dapat bersaing di dunia kerja dengan menggunakan Naïve Bayes: Studi kasus SMK Buddhi Tangerang". Pengujian ini menggunakan confusion matrix dan kurva ROC dengan target berupa belum mampu bersaing dan mampu bersaing. Berdasarkan hasil evaluasi dan validasi yang telah dilakukan, algoritma naïve bayes memiliki akurasi dan performa baik dengan nilai akurasi 98% dan nilai AUC 0,980 [4].

Penelitian yang dilakukan oleh Yufika Septiani, Pipin Farida Ariyani dengan judul "Penerapan Algoritma Naïve Bayes Menentukan Klasifikasi Tingkat Kelulusan Siswa SMK Media Informatika Jakarta" pada tahun 2022. Data yang digunakan adalah data arsip nilai pihak sekolah sebanyak 200 data. Hasil dari penelitian ini mendapatkan perhitungan persentase tingkat akurasi sebesar 90%, recall sebesar 94,44 dan presisi sebesar 94,4% [5].

Penelitian yang dilakukan oleh Esty Purwaningsih, Ela Nurelasari dengan judul "Penerapan K-Nearest Neighbor untuk klasifikasi tingkat kelulusan pada siswa" pada tahun 2021. Data yang diuji menggunakan metode K-Nearest Neighbor dan dataset dibagi menjadi 2 yaitu data training dan data testing menggunakan fold cross validation. Dalam penelitian ini menggunakan rapidminer sebagai pengolahan data. Hasil dari prediksi tingkat kelulusan siswa dengan KNN mendapatkan rata-rata akurasi sebesar 96,49% [6]

Untuk melakukan klasifikasi, penelitian ini menggunakan tiga algoritma machine learning diantaranya adalah Extream Gradient Boosting (XGBoost), Random Forest dan Logistic Regression. *XGBoost* merupakan salah satu algoritma machine learning yang mampu mangatasi permasalahan regresi dan klasifikasi berdasarkan Boosting Decision Tree (GBDT) serta dapat membangun boosted trees secara efisien dan beroperasi secara parallel. Pada dasarnya algoritma ini merupakan metode ensemble yang didasarkan pada gradient boosting tree [7]. Umumnya, metode boosting bekerja dengan cara membuat model baru secara bertahap dengan mempertimbangkan kesalahan model sebelumnya, lalu menambahkan model baru dengan model yang ada. Hal ini dilakukan untuk meningkatkan kompleksitas model akhir [8].

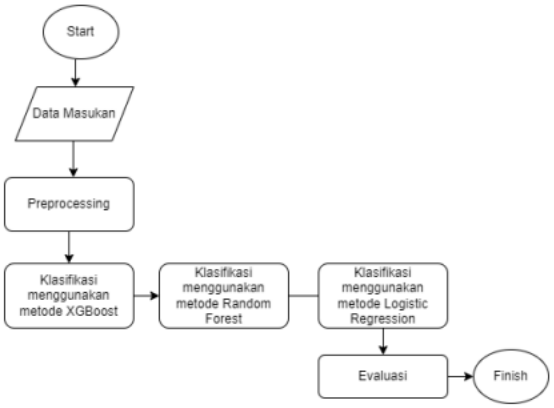
Random Forest merupakan algoritma machine learning dalam decision tree atau pohon keputusan yang digunakan untuk pengklasifikasian dataset dalam jumlah besar atau big data. Algoritma ini digunakan untuk membuat pohon keputusan yang terdiri dari root node, internal node, dan leaf node dengan memilih atribut dan kata secara acak sesuai dengan aturan yang diterapkan. Root node digunakan untuk mengumpulkan data, Internal node yang terdapat di rood node berisi pertanyaan tentang data, sedangkan leaf node digunakan untuk memechkan masalah serta membuat keputusan [9].

Sedangkan Algoritma Logistic Regression merupakan jenis analisis statistic yang sering digunakan untuk pemodelan prediktif. Algoritma ini memprediksi pada saat variable dependen (y) atau output suatu data berupa biner atau berbagai opsi lain [10].

Berdasarkan latar belakang yang telah dipaparkan diatas, penelitian ini mengkaji tentang sebuah sistem yang dapat melakukan klasifikasi lulusan SMK dengan empat kelas yaitu bekerja, belum bekerja, mahasiswa dan wirausaha. Sehingga penelitian ini diharapkan dapat membantu pihak lembaga sekolah untuk melakukan evaluasi dan membuat srategi agar lulusan peserta didik menjadi lebih baik di tahun berikutnya.

II. METODE

Metodologi penelitian merupakan gambaran umum mengenai alur penelitian yang akan dilakukan. Gambar 1. merupakan alur penelitian yang digunakan untuk klasifikasi dari ketiga metode penelitian yang dimulai dari menginputkan data yang sudah diperoleh dan melakukan *preprocessing* untuk membuat model data dan setelah itu melakukan pengklasifikasian oleh ketiga metode yaitu metode XGBoost, Random Forest dan Logistic Regression. Pada tahapan terakhir yaitu mengevaluasi menggunakan confusion matrix untuk pengukuran akurasi dari masing-masing metode.



Gambar 1. Tahapan Penelitian

A. Pengumpulan Data

Data yang diguanakan pada penelitian merupakan data credential yang diambil dari SMK Negeri 1 Bangil. Data tersebut terdiri dari 1126 record dengan beberapa indikator diantaranya adalah tahun kelulusan, jurusan, USBN pengetahuan, USBN keterampilan, rata-rata rapor, NUS dan NS. Sedangkan data status dari tracer study merupakan

target perbandingan hasil klasifikasi. Penelitian ini menggunakan data peserta didik kelas 12 tahun ajaran 2020 dan 2021.

B. Preprocessing

Data mentah tidak dapat dilakukan *pocessing* secara langsung. Perlu adanya tahapan *prepocessing* terlebih dahulu. Tahap *preprocessing* bertujuan untuk memodifikasi data guna meningkatkan kualitas data yang digunakan. Terdapat beberapa sub tahapan pada tahap *prepocessing*, diantaranya adalah encoding categorical data, handling *imbalanced* data, normalisasi data dengan minmax scaler.

a. Encoding Categorical Data

Encoding Categorical Data merupakan tahapan yang harus dilakukan jika data yang digunakan terdapat data yang berjenis kategorik. Pada penelitian ini, peneliti menggunakan teknik one hot encoder untuk melakukan encoding data nominal. Data nominal merupakan data yang menghasilkan satu jenis kategori saja. Selain itu, peneliti juga melakukan teknik label encoder untuk melakukan encoding data ordinal. Data ordinal merupakan data yang memiliki urutan atau tingkatan tertentu. Biasanya urutan ini dapat berupa dari yang terendah hingga tertinggi atau sebaliknya [11].

b. SMOTE

Ketika melakukan klasifikasi data, salah satu masalah yang sering terjadi adalah ketidakseimbangan jumlah data antara kelas yang berbeda. Apabila ketidakseimbangan tersebut sangat ekstrim, maka masalah ini dikenal dengan sebutan *rare* atau *imbalanced* data [7].

Synthetic Minority Oversampling Technique (SMOTE) merupakan metode yang diterapkan untuk menangani ketidakseimbangan kelas. Teknik ini bekerja dengan cara menyeimbangkan jumlah distribusi data sampel pada kelas minoritas dengan menyeleksi data sampel tersebut sampai jumlah data sample menjadi seimbang dengan jumlah sampel pada kelas mayoritas [12].

c. Normalisasi Data

Beberapa dataset terdapat perbedaan rentang nilai pada setiap atribut. Perbedaan ini dapat mengakibatkan atribut yang memiliki nilai jauh lebih kecil dibandingkan dengan atribut lainnya menjadi tidak berfungsi. Oleh karena itu, diperlukan *transformasi* data dengan melakukan normalisasi agar rentang nilai pada setiap atribut dapat disamakan dengan skala tertentu [13]. Pada penelitian ini peneliti menggunakan tenik minmax scaler dimana teknik ini merupakan teknik normalisasi data yang mengubah nilai numerik dalam dataset ke skala umum tanpa mendistorsi perbedaan dalam rentang nilai. Rentang nilai yang biasa digunakan adalah 0 hingga 1. Bila dituliskan kedalam rumus *minmax scaler* terlihat pada persamaan 1.

$$X\ new = \frac{X\ old - X\ min}{X \max - X\ min} \qquad (1)$$

Dimana :

$X\ new$: Nilai data yang telah dinormalisasi.

$x\ old$: Nilai data pada atribut sebelum dilakukan normalisasi.

$x\ min$: Nilai terkecil pada atribut.

$x\ max$: Nilai terbesar pada atribut.

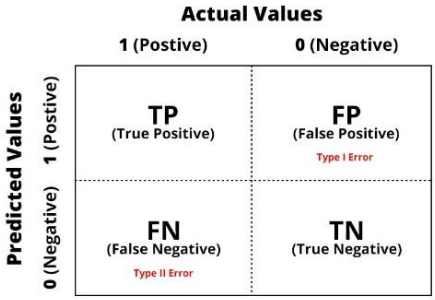
C. Processing

Setelah melalui seluruh tahapan *prepocessing*, tahapan selanjutnya yaitu processing dimana pada tahapan ini dilakukan penerapan algoritma machine learning. Sebelum melakukan penerapan algoritma machine learning, peneliti melakukan pembagian data menjadi dua bagian yaitu data train dan data test dengan persentase 80% untuk data train dan 20% untuk data test.

Untuk mendapatkan parameter yang paling optimal, peneliti menggunakan teknik randomized search cross validation dimana teknik ini dapat melakukan *hyperparameter* tuning lebih cepat dibandingkan grid search cross validation [14]. Setelah mendapatkan parameter terbaik, selanjutnya dilakukkan fitting model dengan tiga algoritma machine learning diantaranya yaitu XGBoost, Random Forest dan Logistic Regression.

D. Evaluasi

Pada tahapan evaluasi, peneliti menggunakan confusion matrix untuk mengukur performa model yang telah dibuat. Confusion matrix berbentuk tabel matrix yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang sebenarnya diketahui. Confusion matrix digunakan untuk membandingkan kelas prediksi dengan kelas data yang sebenarnya [15]



Gambar 1. Confusion Matrix

Pada Gambar 2. Pengukuran confusion matrik menggunakan rumus *accuracy*, *recall*, *precision* dan *micro average F1-score*.

1. *Accuracy* digunakan untuk mengevaluasi performa dari suatu model atau algoritma dalam melakukan prediksi atau klasifikasi. Berikut merupakan rumus *accuracy* dalam persamaan 2 :
   $Accuracy = \frac{TP+TN}{TP+FN+FP+TN}$ (2)

2. *Recall* digunakan untuk mengevaluasi seberapa baik ketepatan jumlah prediksi terhadap suatu kelas yang berhasil diprediksi dengan benar dari total jumlah data yang memiliki label kelas tersebut. Berikut merupakan rumus *recall* dalam persamaan 3 :
   $Recall = \frac{TP}{TP+FN}$ (3)

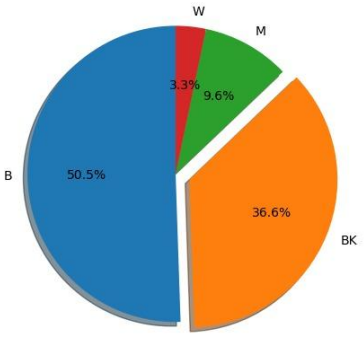
3. *Precision* digunakan untuk mengevaluasi seberapa baik ketepatan jumlah prediksi dengan benar. Berikut merupakan rumus *precision* dalam persamaan 4 :
   $Precission = \frac{TP}{TP+Fp}$ (4)

4. *Micro Average F1-Score* adalah kombinasi pengukuran terhadap *precision* dan *recall* yang digunakan untuk mengukur kombinasi nilai yang telah dihasilkan. Berikut merupakan rumus *F1-score* dalam persamaan 5 :
   $F1 - score = 2\ X\ \frac{Prec\ x\ Rec}{Prec\ x\ Rec}$ (5)

III. Hasil dan Pembahasan

A. Dataset Analysis

Sebelum mengawali proses machine learning, langkah awal yang perlu dilakukan adalah dengan menganalisis dataset yang akan digunakan. Hal ini akan membantu memahami karakteristik dan kualitas data. Langkah pertama yang dilakukan pada penelitian ini yaitu mengelompokkan data perjurusan dengan target sehingga memudahkan untuk mengetahui perbandingan kelas tiap jurusan. Selanjutnya pengelompokan semua jurusan pada tahun 2020-2021 untuk dibandingkan dengan target sehingga menghasilkan persentase perbandingan kelas.



Gambar 2. Visualisasi Perbandingan Dataset

Pada Gambar 3. Menunjukkan bahwa lulusan yang bekerja mendapatkan persentase paling tinggi yaitu 50.5%, belum bekerja mendapat persentase 36.6%, mahasiswa mendapat persentase 9.6% dan wirausaha mendapatkan persentase 3.3%.

B. Data Preprocessing

Pada tahapan *prepocessing* atau modelling data diantaranya yaitu :

One Hot Encoder dan Label Encoder

Data jurusan merupakan data kategorikal, pada *preprocessing* ini akan digunakan teknik one hot encoder untuk mengubah data kategorikal menjadi vector biner atau one-hot, dimana masing-masing fitur hanya dapat bernilai 0 atau 1 dan hanya satu fitur yang bernilai 1 untuk setiap data.

Tabel 1. One Hot Encoder

Status	MM	PSPT	RPL	TB
B	1	0	0	0
BK	1	0	0	0
BK	1	0	0	0
B	0	1	0	0
B	0	1	0	0
M	0	1	0	0
B	0	0	1	0
W	0	0	1	0
W	0	0	0	1
B	0	0	0	1
BK	0	0	0	1

Tabel 2. Menunjukkan bahwa kolom jurusan MM bernilai ‘1’ maka jurusan lainnya akan bernilai ‘0’, begitupun seterusnya.

Data status yang awalnya merupakan data katagorikal, pada *preprocessing* data digunakan teknik label encoder untuk mengubah data katagorikal menjadi angka yang unik. Dikarenakan data status ini memiliki jenjang maka data status di ubah menjadi data ordinal.

Tabel 2. Label Encoder

Status	MM	PSPT	RPL	TB
0	1	0	0	0
1	1	0	0	0
1	1	0	0	0
0	0	1	0	0
0	0	1	0	0
2	0	1	0	0
0	0	0	1	0
3	0	0	1	0
3	0	0	0	1
0	0	0	0	1
1	0	0	0	1

Pada Tabel 3. Menunjukkan data status ‘bekerja’ atau ‘B’ di ubah menjadi ‘0’, status ‘belum bekerja’ atau ‘BK’ menjadi ‘1’, status ‘mahasiswa’ atau ‘M’ menjadi ‘2’, dan status ‘wirausaha’ atau ‘W’ menjadi ‘3’.

Handling Imbalanced Data

Berdasarkan Gambar 3. Visualisasi perbandingan dataset dapat diketahui bahwa terdapat ketidakseimbangan jumlah kelas, artinya data tidak seimbang atau terdapat suatu kondisi dimana sebuah himpunan data terdapat satu kelas

yang memiliki jumlah kecil dibandingkan dengan jumlah kelas yang lain. Oleh karena itu diperlukan adanya tahapan handling imbalanced data.

Tahapan ini melakukan SMOTE dimulai dengan menghitung jarak antara data pada data minoritas, kemudian menentukan nilai persentase SMOTE dan terakhir menetapkan jumlah data terdekat [12].

Tabel 3. Perbandingan Imbalanced Data

Before Oversampling	Counter :	({0:569,1:412, 2:108,3:37})
After Oversampling	Counter :	({0:569,1:569, 2:569,3:569})

Pada Tabel 4. Menunjukkan setelah dilakukan handling *imbalanced* data dengan SMOTE, jumlah data pada tiap kelas menjadi sama, sehingga mengatasi ketidakseimbangan jumlah data pada tiap kelas sebelumnya.

MinMax Scaler

Tabel 4. Hasil Minmax Scaler

TK	UP	UK	RR	NU	NS
0	1	0.7	0.7	0.6	0.6
0	0.3	0.8	0.7	0.7	0.8
1	0.7	0.5	0.6	0.6	0.6
0	1	0.8	0.7	0.6	0.7

Tabel 5. menunjukkan bahwa dalam proses normalisasi, nilai x old dikurangi dengan nilai x min, hasilnya dibagi dengan selisih nilai x max dan x min. Hal ini menghasilkan nilai baru yang berada pada rentang 0 hingga 1, sehingga nilai-nilai pada atribut memiliki skala yang sama dan dapat dibandingkan dengan mudah.

C. Klasifikasi Algoritma

Sebelum dilakukan klasifikasi, dilakukan pembagian data antara data X dan data Y yang mana data X merupakan kolom fitur dan data Y merupakan kolom target. Setelah melakukan pembagian data X dan Y, dilakukan pembagian data train dan data test pada data X menggunakan modul scikit-learn yaitu train_test_split.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.20, random_state=5)
```

Gambar 3. Pembagian dataset

Pada Gambar 4. Pembagian data dibagi menjadi 2 dengan persentase data train 80% dan data test 20%. Selanjutnya melakukan modelling menggunakan algoritma XGBoost, Random Forest dan Logistic Regression dengan parameter tuning.

Tabel 5. XGBoost Hyperparameter Tuning

Hyperparameter	Value
algo__max_depth	Integer(low=1, high=10)
algo__learning_rate	Real(low=-2, high=0, prior='log-uniform')
algo__n_estimators	Integer(low=100, high=200)
algo__subsample	Real(low=0.3, high=0.8, prior='uniform')
algo__gamma	Integer(low=1, high=10)
algo__colsample_bytree	Real(low=0.1, high=1, prior='uniform')
algo__reg_alpha	Real(low=-3, high=1, prior='log-uniform')

algo__reg_lambda	Real(low=-3, high=1, prior='log-uniform')

Tabel 6. Random Forest Hyperparameter Tuning

Hyperparameter	Value
algo__n_estimators	Integer(low=100, high=200)
algo__max_depth	Integer(low=20, high=80)
algo__max_features	Real(low=0.1, high=1, prior='uniform')
algo__min_samples_leaf	Integer(low=1, high=20)

Tabel 7. Logistic Regression Hyperparameter Tuning

Hyperparameter	Value
algo__fit_intercept	[True, False]
algo__C	Real(low=-3, high=3, prior='log-uniform')

Pada Tabel 6, Tabel 7 dan Tabel 8. Merupakan parameter tuning yang disediakan oleh *library jcompl* dengan mengimport fungsi *random_search_params.*

Randomized Search Cross Validation adalah bagian dari modul scikit-learn yang bertujuan secara otomatis dan sistematis melakukan validasi beberapa model dan setiap *hyperparameter*. Ketika proses *running* randomized search cross validation sudah selesai, maka akan didapatkan model beserta skor train dan skor test. Proses tuning dan modelling ditunjukkan pada tabel 9, tabel 10 dan tabel 11 dibawah ini :

Tabel 8. Proses Tuning dan Modelling Metode XGBoost

```
numerical_pipline = Pipeline([
    ('scaler', MinMaxScaler())
])
prepocessor = ColumnTransformer([
    ('numeric', numerical_pipline, X_train.columns)
])
pipeline = Pipeline([
    ('prep', prepocessor),
    ('algo', XGBClassifier())
])
model = RandomizedSearchCV(pipeline, rsp.xgb_params, cv=5, n_iter = 50, n_jobs=-1, verbose=-1, random_state=42)
model.fit(X_train, y_train)
print(model.score(X_train, y_train),model.best_score_,model.score(X_test, y_test))
print(model.score(X_train,y_train)*100, model.score(X_test, y_test)*100)
```

Tabel 9. Proses Tuning dan Modelling Metode Random Forest

```
numerical_pipline = Pipeline([
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', MinMaxScaler())
])
prepocessor = ColumnTransformer([
    ('numeric', numerical_pipline, X_train.columns)
])
pipeline = Pipeline([
```

```
    ('prep', prepocessor),
    ('algo', RandomForestClassifier())
])
model = RandomizedSearchCV(pipeline, rsp.rf_params, cv=5, n_iter = 50, n_jobs=-1, verbose=-1, random_state=42)
model.fit(X_train, y_train)
print(model.score(X_train, y_train),model.best_score_,model.score(X_test, y_test))
print(model.score(X_train,y_train)*100, model.score(X_test, y_test)*100)
```

Tabel 10. Proses Tuning dan Modelling Metode Logistic Regression

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=5)
numerical_pipline = Pipeline([
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', MinMaxScaler())
])
prepocessor = ColumnTransformer([
    ('numeric', numerical_pipline, X_train.columns)
])
pipeline = Pipeline([
    ('prep', prepocessor),
    ('algo', LogisticRegression())
])
model = RandomizedSearchCV(pipeline, rsp.rf_params, cv=5, n_iter = 50, n_jobs=-1, verbose=-1, random_state=42)
model.fit(X_train, y_train)
print(model.score(X_train, y_train),model.best_score_,model.score(X_test, y_test))
print(model.score(X_train,y_train)*100, model.score(X_test, y_test)*100)
```

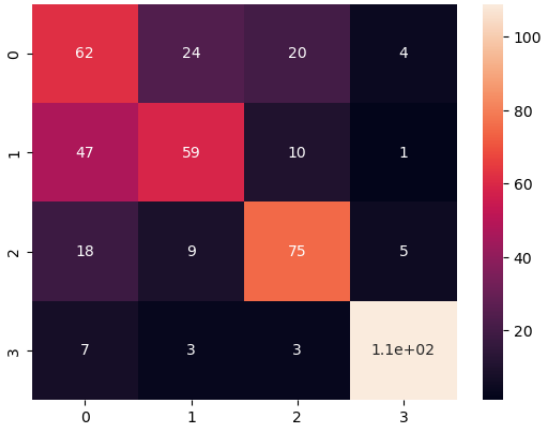
Tabel 11. Perbandingan Best Skor

Algoritma	**Score Train**	**Score Test**
XGBoost	91.70%	66.88%
Random Forest	97.36%	68.71%
Logistic Regression	51.14%	50.43%

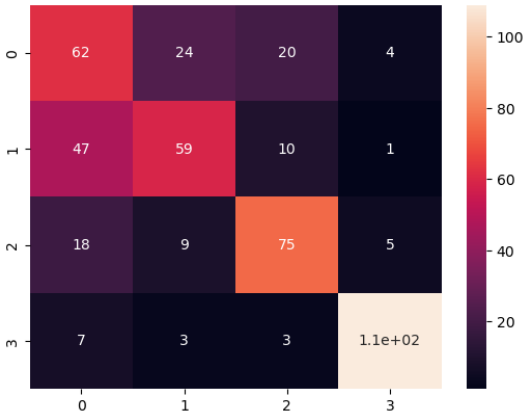
Berdasarkan Tabel 12. Dapat diketahui bahwa terdapat skor test yang paling tinggi yaitu 68.71% yang dihasilkan oleh algoritma Random Forest. Namun dari hasil tersebut mengalami overfitting dengan gap antara skor test dan skor train yang tinggi yaitu 28.65%.

D. Evaluasi

Setelah proses klasifikasi selesai, dilakukan evaluasi model untuk mengukur performa model. Dalam hal ini peneliti menggunakan confusion matrix.



Gambar 4. Confusion Matrix Algoritma XGBoost



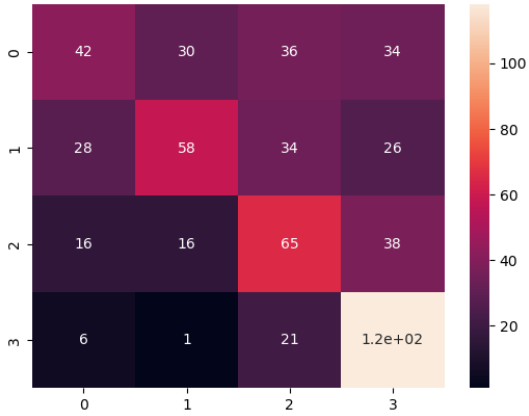
Gambar 5. Confusion Matrix Algoritma Random Forest

Pada Gambar 5 dan Gambar 6. merupakan hasil dari pengukuran model cunfusion matrix pada algoritma XGBoost dan Random Forest. Kedua algoritma tersebut mendapatkan hasil pengukuran model yang sama. Untuk kelas 0 terdapat 62 data yang diklasifikasikan dengan benar, 24 data diklasifikasikan sebagai kelas 1, 20 data diklasifikasikan sebagai kelas 2 dan 4 data diklasifikasikan sebagai kelas 3

Untuk kelas 1 terdapat 59 data yang diklasifikasikan dengan benar, 47 data diklasifikasikan sebagai kelas 0, 10 data diklasifikasikan sebagai kelas 2, dan 1 data diklasifikasikan sebagai kelas 3.

Untuk kelas 2 terdapat 75 data yang diklasifikasikan dengan benar, 18 data diklasifikasikan sebagai kelas 0, 9 data diklasifikasian sebgai kelas 1 dan 5 data diklasifikasikan sebagai kelas 3.

Untuk kelas 3 terdapat 110 data yang diklasifikasikan dengan benar , 7 data diklasifikasikan sebagai kelas 0, 3 data diklasifikasikan sebagai kelas 1 dan 3 data diklasifikasikan sebagai kelas 2.



Gambar 6. Confusion Matrix Algoritma Logistic Regression

Pada Gambar 7. merupakan hasil dari pengukuran model cunfusion matrix pada algoritma Logistic Regression. Untuk kelas 0 terdapat 42 data yang diklasifikasikan dengan benar, 30 data diklasifikasikan sebagai kelas 1, 36 data diklasifikasikan sebagai kelas 2 dan 34 data diklasifikasikan sebagai kelas 3

Untuk kelas 1 terdapat 58 data yang diklasifikasikan dengan benar, 28 data diklasifikasikan sebagai kelas 0, 34 data diklasifikasikan sebagai kelas 2, dan 26 data diklasifikasikan sebagai kelas 3.
Untuk kelas 2 terdapat 65 data yang diklasifikasikan dengan benar, 16 data diklasifikasikan sebagai kelas 0, 16 data diklasifikasian sebagai kelas 1 dan 38 data diklasifikasikan sebagai kelas 3.
Untuk kelas 3 terdapat 120 data yang diklasifikasikan dengan benar , 6 data diklasifikasikan sebagai kelas 0, 1 data diklasifikasikan sebagai kelas 1 dan 21 data diklasifikasikan sebagai kelas 2.

Tabel 12. Evaluasi Perbandingan Confusion Matrix

Algoritma	Accuracy	precision	recall	F1 score
XGBoost	67%	67%	67%	67%
Random Forest	67%	67%	67%	67%
Logistic Regression	50%	50%	50%	50%

Pada tabel 13. dapat diketahui hasil dari pengukuran tiga metode machine learning dalam confusion matrix skor akurasi, precision, recall dan F1-score paling tinggi yaitu 67% yang diperoleh dari algoritma XGBoost dan Random Forest, sedangkan untuk algoritma Logistic Regression mendapatkan skor 50% untuk skor akurasi, precission, recall dan F1-score.

VII. SIMPULAN

Berdasarkan penelitian yang sudah dilakukan, maka dapat ditarik kesimpulan bahwa :
1. Teknik SMOTE membantu mengatasi permasalah pada ketidakseimbangan kelas yang terjadi saat *prepocessing.*
2. Randomized Search Cross Validation bekerja dengan baik untuk menemukan parameter terbaik dari ketiga metode penelitian.
3. Pengujian dilakukan menggunakan confusion matrix dengan dataset sebanyak 1126 dengan pembagian data train 80% dan data test 20%.
4. Algoritma XGBoost dan Random Forest mendapatkan hasil akurasi yang sama yaitu 67%. sedangkan Logistic Regression menghasilkan skor akurasi 50%. dari ketiga metode ini mengalami overfitting.

UCAPAN TERIMA KASIH

Puji dan syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, kami dapat menyelesaikan penelitian ini. Peneliti juga mengucapkan terimakasih kepada Prodi Informatika Universitas Muhammadiyah Sidoarjo yang telah memberikan bimbingan dan fasilitas yang dibutuhkan dalam melakukan penelitian ini. Peneliti juga ingin berterimakasih kepada BRIN yang telah memberikan dukungan dana untuk melakukan penelitian ini. Dan peneliti mengucapkan terimakasih kepada SMKN 1 Bangil yang telah memberikan kesempatan dan membantu dalam melakukan pengumpulan data untuk penelitian ini.

REFERENSI

[1] Y. Sangsurya, M. Muazza, and R. Rahman, "Perencanaan Sumber Daya Manusia Dalam Peningkatan Mutu Pendidikan Di Sd Islam Mutiara Al Madan Kota Sungai Penuh," *J. Manaj. Pendidik. Dan Ilmu Sos.*, vol. 2, no. 2, pp. 766–778, 2021, doi: 10.38035/jmpis.v2i2.644.

[2] S. A. Nurfatimah, S. Hasna, and D. Rostika, "Membangun Kualitas Pendidikan di Indonesia dalam Mewujudkan Program Sustainable Development Goals (SDGs)," *J. Basicedu*, vol. 6, no. 4, pp. 6145–6154, 2022, doi: 10.31004/basicedu.v6i4.3183.

[3] H. Priyono, R. Sari, and T. Mardiana, "Klasifikasi Pemilihan Jurusan Sekolah Menengah Kejuruan Menggunakan Gradient Boosting Classifier," *J. Inform.*, vol. 9, no. 2, pp. 131–139, 2022, doi: 10.31294/inf.v9i2.12654.

[4] L. W. Kusuma, "Prediksi Kemampuan Lulusan SMK untuk Dapat Bersaing Di Dunia Kerja dengan Menggunakan Naïve Bayes : Studi Kasus SMK Buddhi Tangerang," *Prediksi Kemamp. Lulusan SMK untuk Dapat Bersaing Di Dunia Kerja dengan Menggunakan Naïve Bayes Stud. Kasus SMK Buddhi Tangerang*, vol. 1, pp. 56–63, 2019.

[5] Y. Septiani and P. F. Ariyani, "Penerapan Algoritma Naive Bayes Menentukan Klasifikasi Tingkat Kelulusan Siswa SMK Media Informatika Jakarta Application of The Naive Bayes Algorithm Determining

Classification of Students ' Graduation Level of Jakarta Media Informatika Vocational School," no. September, pp. 607–613, 2022.
[6] E. Purwaningsih and E. Nurelasari, "Penerapan K-Nearest Neighbor Untuk Klasifikasi Tingkat Kelulusan Pada Siswa," *Syntax J. Inform.*, vol. 10, no. 01, pp. 46–56, 2021, doi: 10.35706/syji.v10i01.5173.
[7] I. Muslim and K. Karo, "Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan," *J. Softw. Eng. Inf. Commun. Technol.*, vol. 1, no. 1, pp. 10–16, 2020.
[8] M. Rizky Mubarok, Muliadi, and R. Herteno, "Hyper-Parameter Tuning pada XGBoost Untuk Prediksi Keberlangsungan Hidup Pasien Gagal Jantung," *Kumpul. J. Ilmu Komput.*, vol. 9, no. 2, pp. 391–401, 2022.
[9] P. R. Sihombing and I. F. Yuliati, "Penerapan Metode Machine Learning dalam Klasifikasi Risiko Kejadian Berat Badan Lahir Rendah di Indonesia," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 20, no. 2, pp. 417–426, 2021, doi: 10.30812/matrik.v20i2.1174.
[10] A. K. Santoso, A. Noviriandini, A. Kurniasih, B. D. Wicaksono, and A. Nuryanto, "Klasifikasi Persepsi Pengguna Twitter Terhadap Kasus Covid-19 Menggunakan Metode Logistic Regression," *JIK (Jurnal Inform. dan Komputer)*, vol. 5, no. 2, pp. 234–241, 2021.
[11] A. Tjalla and M. Mahdiyah, "Data Kategorik dalam Penelitian : Review Bibliometrik," vol. 9, no. 1, pp. 796–802, 2023, doi: 10.58258/jime.v9i1.4814/http.
[12] A. N. Kasanah, M. Muladi, and U. Pujianto, "Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 196–201, 2019, doi: 10.29207/resti.v3i2.945.
[13] D. A. Nasution, H. H. Khotimah, and N. Chamidah, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN," *Comput. Eng. Sci. Syst. J.*, vol. 4, no. 1, p. 78, 2019, doi: 10.24114/cess.v4i1.11458.
[14] E. Agustin, A. Eviyanti, and N. L. Azizah, "Deteksi Penyakit Epilepsi Melalui Sinyal EEG Menggunakan Metode DWT dan Extreme Gradient Boosting," vol. 7, pp. 117–127, 2023, doi: 10.30865/mib.v7i1.5412.
[15] M. Noveanto, H. Sastypratiwi, and H. Muhardi, "Uji Akurasi Klasifikasi Emosi Pada Lirik Lagu Bahasa Indonesia Emotion Classification Accuracy Test in Indonesian Song Lyrics," vol. 10, no. 3, pp. 311–318, 2022, doi: 10.26418/justin.v10i3.56804.

Conflict of Interest Statement:
The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.