

Prediksi Potensi Penyakit Stroke menggunakan Algoritma *Support Vector Machine*, *Random Forest Classifier*, dan *Logistic Regression*, serta *Artificial Neural Network* Melalui Keras

Leonard Aldo, Fiona Tanadi, Abdul Razak Danish

Sistem Informasi, Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara, Tangerang, Banten 15111

Corresponding author: First A. Author (e-mail: author@boulder.nist.gov).

This paragraph of the first footnote will contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456."

ABSTRACT Penyakit *Stroke* merupakan salah satu penyakit yang dapat menyebabkan kecacatan hingga kematian yang signifikan di seluruh dunia. Kecacatan yang dimaksud adalah kecacatan fisik, kognitif, dan emosional. Pengobatan untuk penyakit *stroke* berupa rehabilitasi juga memerlukan biaya yang tinggi dan waktu yang lama. Oleh karena itu, prediksi penyakit *stroke* sangat penting untuk mencegah dan pengelolaan resiko munculnya penyakit *stroke* sejak dini. Berbagai teknik dan algoritma *machine learning* telah diterapkan dalam membangun model prediksi yang dapat mengidentifikasi kemungkinan terjadinya *stroke* pada seorang individu. Metode penelitian yang digunakan adalah CRISP-DM. Algoritma *machine learning* yang digunakan sebagai model untuk melakukan prediksi dalam penelitian ini adalah *Random Forest*, *Support Vector Machine (SVM)* *Logistic Regression*, dan *Artificial Neural Network* melalui library *Tensorflow Keras*. Melalui hasil pembuatan model untuk prediksi penyakit *stroke*, setelah dilakukan *hyperparameter tuning*, didapatkan nilai akurasi sebesar 86.07% dan skor F1 15.30% untuk model *Random Forest*, akurasi 69.89% dan skor F1 16.44% untuk model *Support Vector Machine (SVM)*, akurasi 73.63% dan skor F1 20.55% untuk model *Logistic Regression*, dan akurasi 72.49% serta skor F1 19% untuk model *Artificial Neural Network*. Dari keempat model tersebut, model yang dipilih untuk *deployment* adalah model *Logistic Regression* dan *Artificial Neural Network*, dikarenakan performa kedua model pada nilai *recall*, *precision*, dan skor F1 yang baik.

INDEX TERMS *Artificial Neural Network*, *CRISP-DM*, *Logistic Regression*, *Random Forest*, *Stroke*, *Support Vector Machine*.

I. INTRODUCTION

Stroke merupakan penyakit berbahaya yang dapat menyebabkan kematian. Menurut data WHO pada tahun 2019, penyakit *stroke* menduduki peringkat pertama dari 10 penyakit penyebab kematian tertinggi di Indonesia, dengan jumlah 131,8 kasus kematian per seratus ribu penduduk, disusul dengan 95,68 kasus serangan jantung di posisi kedua dan 40,78 kasus diabetes melitus di posisi ketiga [1]. Penyakit *stroke* umumnya disebabkan oleh tekanan darah tinggi, obesitas, stress, faktor genetik dan depresi berlebihan. Untuk mencegah penyakit *stroke*, dapat dilakukan dengan cara menghilangkan kebiasaan buruk seperti merokok dan minum minuman beralkohol [2].

Deteksi penyakit *stroke* sejak dini bermanfaat untuk meminimalisir dampak penyakit *stroke* pada bagian otak maupun bagian tubuh lainnya. Dengan adanya prediksi penyakit *stroke*, dapat dilakukan pengobatan dini untuk mencegah penyakit *stroke*, memperbesar peluang penyembuhan dan mengurangi tingkat kematian [3]. Prediksi *stroke* dapat dilakukan dengan bantuan *machine learning* melalui algoritma prediksi. Penggunaan *machine learning* terbukti banyak diterapkan dalam topik klasifikasi dan optimasi dalam pembuatan sistem cerdas dalam peningkatan penyedia layanan kesehatan [4] [5]. Algoritma *machine learning* yang digunakan dalam penelitian ini antara lain *Support Vector Machine (SVM)*, *Random Forest*, *Logistic Regression*, dan *Artificial Neural Network*.

Keempat algoritma tersebut akan dikomparasi untuk mencari model terbaik dalam melakukan prediksi untuk tahap *deployment*.

II. LITERATURE REVIEW

Prediksi penyakit stroke merupakan sebuah solusi yang bermanfaat dalam membantu pihak medis mendiagnosa potensi penyakit stroke yang dimiliki seorang individu. salah satu penelitian serupa adalah penelitian yang dilakukan oleh Amelia et al. (2022), yang menggunakan algoritma SVM untuk memprediksi penyakit stroke sesuai dengan atribut berpengaruh dengan akurasi yang tinggi yaitu 100% [6].

Penelitian lain yang membandingkan metode data mining untuk prediksi stroke dilakukan oleh Airi et al. (2023) yang membandingkan algoritma *Naïve Bayes*, *K-Nearest Neighbor* dan *Random Forest*, dengan akurasi sebesar 71.9% untuk *Naïve Bayes*, 73.6% untuk *K-Nearest Neighbor* dan sebesar 92.5% untuk *Random Forest* [7]. Sementara itu, implementasi *library Keras* yang digunakan dalam penelitian ini telah terkenal digunakan untuk banyak penelitian pembelajaran mesin, salah satunya penelitian yang dilakukan oleh Cheon, et al. (2019), yang mengimplementasikan pendekatan DNN (*deep neural network*) dengan penggunaan *library tensorflow Keras* [8].

Pemilihan algoritma SVM sebagai salah satu algoritma dalam penelitian ini merujuk pada penelitian sebelumnya yang menggunakan algoritma *Support Vector Machine* (SVM) yang dilakukan oleh Sulaeman et. al (2022) menjelaskan proses pengujian algoritma SVM untuk klasifikasi *stroke* dengan menggunakan data *kernel linear unbalanced polinomial*, memperoleh akurasi sebesar 80% [9]. Penelitian serupa yang dilakukan oleh Handayani (2021) membandingkan hasil klasifikasi penyakit jantung dengan tiga algoritma *machine learning* yang berbeda yaitu *Support Vector Machine (SVM)*, *Logistic Regression*, dan *Artificial Neural Network*. Pada penelitian tersebut, model *Logistic Regression* memperoleh akurasi tertinggi sebesar 86% dengan skenario pembagian data 80:20 [10].

Dikarenakan data yang diperoleh untuk penelitian ini merupakan data dengan kategori yang tidak seimbang, digunakan metode *Synthetic Minority Oversampling Technique (SMOTE)*. Metode *SMOTE* telah terbukti mengatasi masalah data yang tidak seimbang dan dapat memberikan hasil yang optimal dan peningkatan akurasi, seperti penelitian yang dilakukan oleh Hidayat et. al (2021) tentang pengaruh *SMOTE* pada kinerja SVM terhadap kumpulan data Airbnb yang tidak seimbang. Didapatkan peningkatan akurasi dari 80% menjadi 81% karena dengan *SMOTE*, muncul pengaruh peningkatan hasil uji *true label* (minoritas) dan hasil uji *false label* (mayoritas) [11]. Selain itu, penelitian lain yang menerapkan *SMOTE* yaitu penelitian oleh Stoyo (2020), yang menguji kinerja ANN + *SMOTE* memiliki akurasi sebesar 87.06% jika dibandingkan dengan ANN (*Artificial Neural Network*) yang hanya sebesar 86.35% [12].

Keempat algoritma yang dipilih untuk dikomparasi dalam penelitian ini merupakan model algoritma yang optimal dan memiliki tingkat akurasi yang baik dan telah banyak digunakan dalam beberapa penelitian sebelumnya, salah satunya penelitian yang dilakukan oleh Sailasya et al. (2021) dengan menggunakan *Random Forest*, *Logistic Regression*, *Support Vector Machine* dan 3 algoritma *machine learning* lainnya dalam menganalisa performa prediksi stroke memperoleh akurasi masing-masing sebesar 78% untuk *Logistic Regression*, 80% untuk *Support Vector Machine*, dan 73% untuk *Random Forest* [13].

Tujuan dari penelitian ini adalah untuk mengimplementasikan model *machine learning* untuk memprediksi pasien dengan potensi penyakit stroke. Dataset yang digunakan adalah data medis pasien dengan parameter *gender*, *age*, rekam medis penyakit hati, dan status merokok yang bersumber dari *website Kaggle*.

III. METHODOLOGY

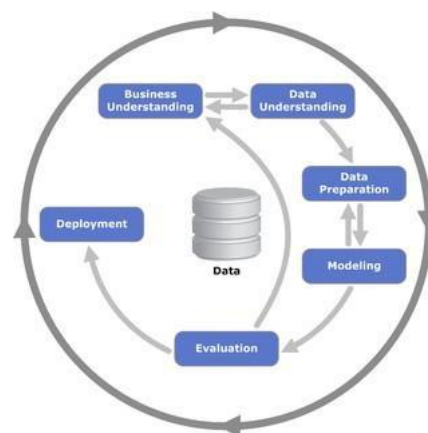


FIGURE 1. CRISP-DM Methodology

Metode yang digunakan dalam penelitian ini adalah CRISP-DM (*Cross Industry Standard Process for Data Mining*). CRISP-DM merupakan sebuah metode terkenal yang sering digunakan dalam *project data science* [14]. Metode ini terbagi menjadi enam tahapan utama, yaitu *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*. Proses tersebut masing-masing dijabarkan sebagai berikut.

A. BUSINESS UNDERSTANDING

Business Understanding merupakan pemahaman yang dibutuhkan terhadap solusi yang dibutuhkan untuk menyelesaikan masalah, dalam rangka memaksimalkan efisiensi dari mesin untuk mencapai *goals* tertentu dalam permasalahan *data mining* [15].

Business understanding atau pemahaman bisnis dalam penelitian ini yaitu mengidentifikasi apakah seorang individu memiliki potensi terkena stroke atau tidak dengan adanya data tentang individu tersebut, sehingga tindakan pencegahan dapat disarankan dan resiko terkena penyakit stroke dari seorang individu dapat diminimalisir.

Mengingat, potensi ataupun resiko terkena penyakit stroke merupakan hal yang cukup krusial dalam kehidupan seorang individu sehingga dibutuhkan pengambilan keputusan yang tepat dalam mencegahnya.

Prediksi penyakit stroke dapat dilakukan melalui aplikasi yang mengimplementasikan model terbaik yang telah di *deploy* dari penelitian ini. Tujuannya, dengan adanya aplikasi yang mudah di akses dan selalu tersedia, penggunaan aplikasi dapat lebih efektif dan efisien.

B. DATA UNDERSTANDING

Setelah mendapatkan gambaran dari permasalahan yang ingin diselesaikan dari *business understanding*, perlu dilakukan pengamatan terhadap data yang akan digunakan dalam penelitian guna memahami informasi yang disediakan oleh data. Dataset yang digunakan merupakan data yang mencakup informasi medis dan data demografis dari 5,110 individu dengan 10 fitur yang berbeda, serta kejadian stroke. Artinya, penelitian akan menggunakan *supervised learning* (Pendekatan dengan data *training* yang sudah memiliki label) [14]. Dengan menggunakan pendekatan ini, algoritma yang dipilih untuk membangun model juga harus sesuai dengan kebutuhan.

C. DATA PREPARATION

Tahapan ketiga dari CRISP-DM yaitu data preparation, dimana dataset dimodifikasi dan di atur sehingga data dalam digunakan dalam implementasi algoritma *machine learning* untuk proses *training*. Pada tahapan ini, data akan dibagi menjadi 2 bagian yaitu data *training* dan data *testing*, sehingga informasi yang dibutuhkan untuk kedua jenis *learning* memadai sehingga model nantinya juga dapat dievaluasi.

D. MODELLING

Beberapa teknik modeling tersedia secara luas dalam dunia data analisis. Oleh karena itu, perlu dipilih algoritma yang paling sesuai dalam pembuatan model sehingga *output* yang dihasilkan dapat memenuhi permasalahan yang ingin diselesaikan. Jika penelitian menggunakan lebih dari 1 model, maka perlu dilakukan evaluasi untuk mengidentifikasi performa dan nilai akurasi yang terbaik untuk menyelesaikan masalah. Untuk tahapan ini, empat buah algoritma *machine learning* yang berbeda diimplementasikan dan dikomparasi untuk mencari model dengan akurasi dan performa terbaik untuk digunakan sebagai model dalam *deployment*.

E. EVALUATION

Setelah melalui tahapan modeling, tahapan berikutnya adalah untuk mengevaluasi performa dari setiap model terhadap dataset. Tidak hanya nilai akurasi, namun nilai lain seperti *recall* dan *precision* juga digunakan sebagai parameter untuk mengevaluasi model.

F. DEPLOYMENT

Tahapan terakhir adalah menggunakan hasil dan informasi yang didapatkan dari penelitian. Artinya, model yang telah selesai di buat harus di *deploy* agar tersedia secara publik, sehingga pebisnis yang tertarik terhadap aplikasi ataupun masyarakat dapat menerima manfaat dari penelitian ini, dan dapat juga digunakan oleh peneliti lain untuk penelitian atau analisa lanjutan berikutnya. Untuk mencapai *deployment*, model terbaik yang didapatkan dari hasil evaluasi disimpan terlebih dahulu sebelum dicoba konfigurasi agar sistem dapat diakses dan tersedia secara *online*.

IV. RESULT AND DISCUSSION

Dataset yang digunakan dalam penelitian ini mencakup informasi dari 5,110 individu dengan 10 fitur utama yang mencakup informasi mengenai variable target Stroke untuk pendekatan *supervised learning*. Seperti yang terlihat pada tabel 1, fitur yang terdapat dalam dataset mencakup data rekam medis dengan *value* atau nilai yang dapat berupa numerikal maupun kategorikal.

TABLE I
DESKRIPSI DATASET

Features	Values
Data Demografis	
Jumlah individu	5110
Umur (tahun)	1-82
Gender	Male/Female
Ever Married	Yes/No
Residence Type	Rural/Urban
Work Type	Children/Govt_job/Never_ Worked/Private/Self-employ ed
Data Medis	
Hypertension	0 (No) /1 Yes
Heart Disesease	0 (No) /1 Yes
Avg Glucose Level	55.12-271.74
BMI (<i>Body Mass Index</i>)	10.3-97.6
Stroke	0 (No) /1 Yes
Smoking Status	Formerly Smoked/Never Smoked/Smokes/Unknown

Hasil dari proses yang telah dilakukan dalam penelitian, dijabarkan sebagai berikut.

A. DATA PREPARATION

Data yang telah diperoleh dari *website Kaggle* kemudian di import ke dalam *notebook* dan dimasukkan ke sebuah *dataframe*. Data kemudian dipersiapkan terlebih dahulu melalui proses *data preparation* sebelum dapat digunakan untuk *training*. Dengan menggunakan *function* info, kita dapat melihat informasi mengenai keseluruhan data, seperti pada figure 2. dataset

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype  
---  --
0   id                     5110 non-null   int64  
1   gender                 5110 non-null   object  
2   age                    5110 non-null   float64
3   hypertension           5110 non-null   int64  
4   heart_disease          5110 non-null   int64  
5   ever_married           5110 non-null   object  
6   work_type              5110 non-null   object  
7   Residence_type         5110 non-null   object  
8   avg_glucose_level      5110 non-null   float64
9   bmi                    4909 non-null   float64
10  smoking_status         5110 non-null   object  
11  stroke                  5110 non-null   int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

FIGURE 2. DATASET INFO

```
df.isna().sum()

id                0
gender            0
age              0
hypertension      0
heart_disease     0
ever_married      0
work_type         0
Residence_type    0
avg_glucose_level 0
bmi              201
smoking_status    0
stroke            0
dtype: int64
```

FIGURE 3. NULL DATA

Berdasarkan figure 3, terlihat bahwa *dataset* sebenarnya memiliki nilai yang kosong (NaN) atau hilang (*missing values*) sebanyak 201 buah pada fitur BMI (*Body Mass Index*). Oleh karena itu, data yang kosong dengan menggunakan median atau mean dengan menggunakan algoritma *decision tree*.

Pada dataset stroke, terdapat 5 fitur dengan tipe data *string*, yaitu data dengan jenis kategorikal (*object*) sehingga perlu diubah menjadi bentuk numerik dengan *Label Encoder* agar data dapat digunakan oleh model. Setelah selesai melalui tahapan *encoding*, korelasi antar fitur dicari dengan melakukan visualisasi *heatmap* seperti pada figure 4. Terlihat bahwa fitur penting yang dapat digunakan adalah fitur *age*, *hypertension*, *heart disease*, dan *average glucose level*. Dataset kemudian dibagi menjadi dua bagian, *training* dan *testing*, dengan komposisi pembagian training adalah sebesar 30%.

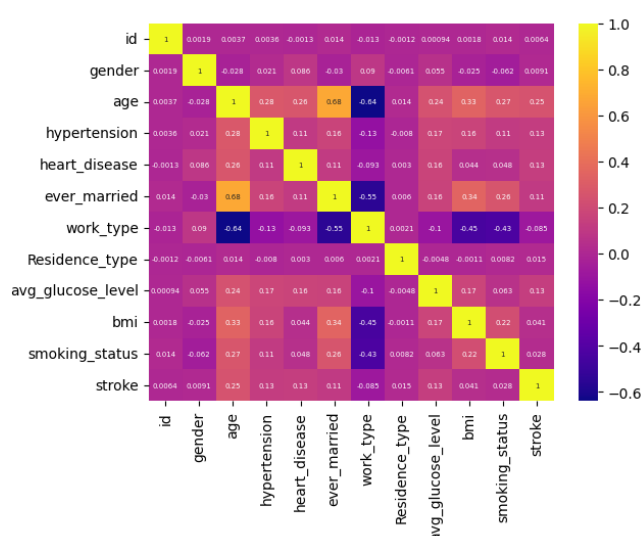


FIGURE 4. FEATURE CORRELATION HEATMAP

B. SMOTE FOR UNBALANCED DATASET

Dari hasil 10 data *training* awal, diketahui bahwa data yang diperoleh merupakan data yang tidak seimbang (*unbalanced*). Oleh karena itu, digunakan metode SMOTE (*Synthetic Minority Oversampling Technique*) untuk mengambil *sample* minoritas untuk menyamakan kedudukan populasi dalam data agar data dapat lebih seimbang. Seperti yang terlihat pada figure 5, bahwa jumlah data kategori 1 (berpotensi stroke) dengan jumlah kategori 0 (tidak berpotensi stroke) sudah seimbang dengan nilai 50% dan 50%.

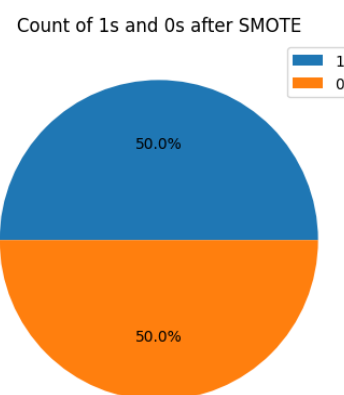


FIGURE 5. COUNT of 1s and 0s after SMOTE

C. MODEL DESIGN

Terdapat empat buah algoritma yang digunakan untuk merancang model, yaitu *Random Forest*, *Support Vector Machine* (SVM), *Logistic Regression*, dan

Artificial Neural Network (ANN). Penjelasan dari masing-masing algoritma dijabarkan sebagai berikut.

1) *RANDOM FOREST CLASSIFIER* (RF)

Algoritma *Random Forest* (RF) merupakan sebuah algoritma yang dapat digunakan untuk membuat model untuk prediksi. Algoritma RF terdiri dari banyak *decision tree* independen yang dilatih secara independent pada data dengan *subset* yang acak (*random*). Pohon-pohon (*trees*) terbentuk pada waktu *training*, dan *output* yang dihasilkan didapat dari setiap *decision tree*. Algoritma RF seringkali menghasilkan akurasi yang besar dibandingkan dengan model tunggal *decision tree* pada umumnya [13]. Salah satu keuntungan menggunakan algoritma RF untuk melakukan prediksi adalah karena kemampuannya untuk menangani dataset dengan jumlah *variable* prediksi yang besar [16].

Model *random forest* dirancang dengan menggunakan *library* *scikit-learn* yaitu *function* *RandomForestClassifier*. Kemudian digunakan beberapa parameter seperti *n_estimators* yang merepresentasikan jumlah *trees* dalam *random forest*, *max_features* menyatakan jumlah maksimal dari fitur untuk setiap *split*, dan *bootstrap* untuk menggunakan *sample bootstrap* untuk membangun *trees*. Hasil akurasi yang didapatkan dengan model *random forest* adalah sebesar 86%.

2) *SUPPORT VECTOR MACHINE* (SVM)

Algoritma *Support Vector Machine* (SVM) adalah sebuah algoritma yang mempelajari model dengan menggunakan fungsi linear dalam suatu fitur yang dilatih dengan algoritma yang didasari oleh teori yang optimal. Prinsip dasar algoritma SVM adalah mengembangkan klasifikasi *linear* agar dapat diproses pada masalah *non-linear*, dengan menggunakan metode *kernel trick* pada fitur dengan dimensi tinggi [6]. Salah satu tipe kernel yang umum digunakan adalah kernel RBF (*Radial Basis Function*), dengan persamaan seperti pada rumus (1) [6].

$$G(x_1, x_2) = \exp\#(1)$$

Model SVM dirancang dengan *function* seperti *svm_param_grid* sebagai *hyperparameter grid* yang digunakan saat *grid search*. Parameter yang di *tune* yaitu 'C' sebagai parameter regularisasi, 'gamma' sebagai koefisien *kernel*, dan *kernel* sebagai tipe fungsi *kernel* yang digunakan, yaitu RBF (*Radial Basis Function*).

3) *LOGISTIC REGRESSION* (LR)

Algoritma *Logistic Regression* (LR) merupakan algoritma dalam *supervised learning* yang sering digunakan dalam memprediksi probabilitas dari sebuah *variable output*. Algoritma LR memiliki nilai

fit terbaik ketika nilai *variable output* nya memiliki nilai biner (0 atau 1) [13].

Model LR dirancang dengan beberapa parameter seperti *penalty* yang merepresentasikan tipe *penalty regularization*, yaitu L1 (*Lasso*) atau L2 (*Ridge*). Kemudian digunakan 'C' sebagai *hyperparameter* yang mengontrol *inverse* dari kekuatan regularisasi, dimana nilai yang kecil merepresentasikan menggambarkan regularisasi yang kuat.

4) *ARTIFICIAL NEURAL NETWORK* (ANN)

Algoritma *Artificial Neural Network* (ANN) merupakan sebuah model komputasional yang struktural dan fungsinya dibuat berdasarkan inspirasi dari *biological neural networks*, seperti otak manusia. Saat melatih model, ANN belajar dari sebuah dataset yang sudah dilabel sebelumnya dengan mengatur *weights* dari koneksi untuk meminimalisir perbedaan antara *predicted outputs* dengan *true outputs*. ANN mampu mempelajari hubungan non-linear dan dapat diaplikasikan pada beberapa *tasks*, seperti prediksi, klasifikasi, *pattern recognition*, dan regresi [17].

Model ANN dirancang dengan menggunakan model *sequential* Keras, dengan 3 layer *dense*, 2 fungsi aktivasi ReLU dan 1 fungsi aktivasi *sigmoid*. Dengan 10 epochs, dan metrik pengukuran adalah akurasi.

D. *MODEL EVALUATION*

Setelah model dirancang, *cross validation* kemudian dilakukan terhadap data yang telah dipersiapkan melalui tahapan *data preparation* sebelumnya. *Cross Validation* berfungsi untuk menentukan *baseline* dari setiap model untuk dibandingkan dengan model yang dibuat dan membantu dalam memilih *hyperparameter* terbaik, serta mengurangi potensi *overfitting* dan menyediakan estimasi performa model yang dapat diandalkan [18].

```
Mean f1 scores:
Random Forest mean : 0.9105956484448574
SVM mean : 0.8234300364960788
Logistic Regression mean : 0.7876405822666073
```

FIGURE 6. MEAN F1 SCORES DATASET TRAIN

Dari hasil model *cross validation data training*, didapatkan skor F1 untuk model *Random Forest* merupakan model dengan nilai F1 tertinggi yaitu sebesar 91.05%, disusul dengan SVM sebesar 82.34%, dan *Logistic Regression* sebesar 78.76%. Skor F1 bermanfaat untuk melihat apakah model yang dibuat *overfit* atau tidak untuk dataset *test*. Adapun rumus dari F1 yaitu seperti yang terlihat pada persamaan (2) [19].

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \#(2)$$

Precision sendiri merupakan pencarian nilai positif dari hasil prediksi yang bernilai positif, untuk membuktikan akurasi model dalam menemukan jawaban yang benar. Adapun persamaan dari *precision* yaitu seperti pada rumus (3) [20].

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \times 100\% \#(3)$$

Sementara itu, *recall* merupakan nilai pencarian nilai positif yang berhasil diprediksi dari keseluruhan total nilai positif sebenarnya. Adapun rumus *recall* seperti pada rumus (4) [20].

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \times 100\% \#(4)$$

Untuk melihat apakah model *overfit* atau tidak, dapat kita lihat dari skor F1 untuk data *test*. Dari hasil *cross validation data testing*, didapatkan skor F1 untuk model algoritma *Logistic Regression* dengan nilai tertinggi yaitu 20.54%, yang artinya algoritma *logistic regression* menjadi algoritma terbaik untuk melakukan prediksi stroke. Nilai F1 yang dimaksud seperti yang terlihat pada *figure 7*.

```
Mean f1 scores:
RF mean : 0.15916955017301038
SVM mean : 0.1842772612003381
LogReg mean : 0.2054794520547945
```

FIGURE 7. MEAN F1 SCORES DATASET TEST

Setelah mendapatkan nilai F1, dapat kita buat *classification report* pada model *random forest* untuk melihat nilai akurasinya.

```
precision    recall  f1-score   support

0           0.96      0.89      0.93      3404
1           0.11      0.27      0.16       173

accuracy          0.86      3577
macro avg         0.54      0.58      0.54      3577
weighted avg      0.92      0.86      0.89      3577

Accuracy Score: 0.8641319541515237
```

FIGURE 8. CLASSIFICATION REPORT OF RF MODEL

Dari *classification report* pada *figure 8*, nilai *recall* yang didapatkan adalah sebesar 0.27, yang berarti model *random forest* hanya bisa menebak sebanyak 27% dari dataset *test* secara benar, meskipun didapatkan nilai akurasi sebesar 86%. Untuk menangani masalah ini, dilakukan *hyperparameter tuning* untuk masing-masing model untuk dibandingkan kembali hasil dan akurasinya.

E. HYPERPARAMETER TUNING & MODEL COMPARISON

Model yang telah selesai dievaluasi kemudian dibandingkan setelah melalui tahapan *hyperparameter tuning*. Adapun hasil dari masing-masing model adalah sebagai berikut.

1) RANDOM FOREST CLASSIFIER (RF)

Untuk melihat akurasi model, dibuat *classification report*. Terlihat pada *figure 9* bahwa nilai akurasi yang didapatkan dari model *Random Forest* adalah sebesar 86.07% setelah melalui *hyperparameter tuning*.

```
precision    recall  f1-score   support

0           0.96      0.89      0.92      3404
1           0.11      0.26      0.15       173

accuracy          0.86      3577
macro avg         0.53      0.58      0.54      3577
weighted avg      0.92      0.86      0.89      3577

Accuracy Score: 0.8607771875873638
```

FIGURE 9. CLASSIFICATION REPORT OF TUNED RF MODEL

Berdasarkan *classification report* pada *figure 9*, dapat disimpulkan bahwa untuk kelas 0, ketepatannya adalah 0,96, yang menunjukkan bahwa dari semua contoh yang diprediksi berlabel 0,96% *True Negative*. Untuk kelas 1, ketepatannya adalah 0,11, menunjukkan bahwa hanya 11% dari contoh yang diprediksi sebagai kelas 1 adalah positif.

Untuk kelas 0, *recall* adalah 0,89, menunjukkan bahwa 89% dari *True Negative* diidentifikasi dengan benar sebagai kelas 0. Untuk kelas 1, *recall* adalah 0,26, menunjukkan bahwa 26% dari *True Positive* diidentifikasi dengan benar sebagai kelas 1.

Untuk kelas 0, skor F1 adalah 0,92, dengan mempertimbangkan keseimbangan antara *precision* dan *recall*. Untuk kelas 1, skor F1 adalah 0,15, yang mengindikasikan kinerja yang lebih rendah karena *precision* yang rendah dan *recall* yang juga rendah.

Kemudian, pada *figure 10* merupakan hasil visualisasi *learning curve* dari model *random forest*. Terlihat bahwa model *random forest* memiliki performa yang kurang baik karena nilai *training* cenderung lebih tinggi dibandingkan dengan performa *cross-validation*, sehingga, model *random forest* mengalami *overfitting*, meskipun telah dilakukan *bias removal* dan *hyperparameter tuning*.

Selain itu, dilihat dari *recall score* yang dimiliki oleh *random forest* yaitu sebesar 26%, algoritma *random forest* kurang memadai untuk *deployment*. Oleh karena itu, akan dicoba dengan melihat evaluasi model tiga algoritma lainnya.

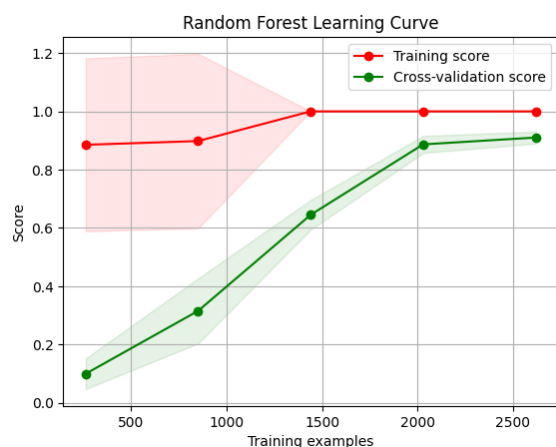


FIGURE 10. RANDOM FOREST LEARNING CURVE

2) SUPPORT VECTOR MACHINE (SVM)

Untuk melihat akurasi model, dibuat *classification report*. Terlihat pada figure 11 bahwa nilai akurasi yang didapatkan dari model SVM adalah sebesar 70% setelah melalui *hyperparameter tuning*.

	precision	recall	f1-score	support
0	0.97	0.70	0.82	3404
1	0.09	0.61	0.16	173
accuracy			0.70	3577
macro avg	0.53	0.66	0.49	3577
weighted avg	0.93	0.70	0.78	3577

Accuracy Score: 0.698909700866648
F1 Score: 0.16446858029480219

FIGURE 11. CLASSIFICATION REPORT OF TUNED SVM MODEL

Berdasarkan *classification report* pada figure 11, dapat disimpulkan bahwa untuk kelas 0, ketepatannya adalah 0,97, yang menunjukkan bahwa dari semua contoh yang diprediksi berlabel 0,97% True Negative. Untuk kelas 1, ketepatannya adalah 0,09, menunjukkan bahwa hanya 9% dari contoh yang diprediksi sebagai kelas 1 adalah positif.

Untuk kelas 0, *recall* adalah 0,70, menunjukkan bahwa 70% dari True Negative diidentifikasi dengan benar sebagai kelas 0. Untuk kelas 1, *recall* adalah 0,61, menunjukkan bahwa 61% dari True Positive diidentifikasi dengan benar sebagai kelas 1.

Untuk kelas 0, skor F1 adalah 0,82, dengan mempertimbangkan keseimbangan antara *precision* dan *recall*. Untuk kelas 1, skor F1 adalah 0,16, yang mengindikasikan kinerja yang lebih rendah karena *precision* yang rendah dan *recall* yang sedang.

Kemudian, pada figure 12 merupakan hasil visualisasi *learning curve* dari model SVM. Terlihat bahwa model SVM memiliki performa yang kurang baik karena nilai *cross-validation* cenderung lebih tinggi dibandingkan nilai *training*, sehingga, model SVM mengalami

underfitting, meskipun telah dilakukan *hyperparameter tuning*.

Selain itu, dilihat dari *recall score* yang dimiliki oleh SVM yaitu sebesar 61%, algoritma SVM kurang memadai untuk *deployment*. Oleh karena itu, akan dicoba dengan melihat evaluasi model dua algoritma lainnya.

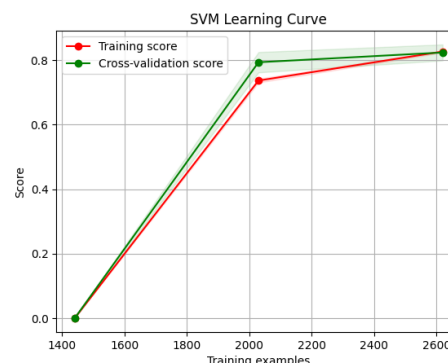


FIGURE 12. SVM LEARNING CURVE

3) LOGISTIC REGRESSION (LR)

Untuk melihat akurasi model, dibuat *classification report*. Terlihat pada figure 13 bahwa nilai akurasi yang didapatkan dari model LR adalah sebesar 73.63% setelah melalui *hyperparameter tuning*.

	precision	recall	f1-score	support
0	0.98	0.74	0.84	3404
1	0.12	0.71	0.21	173
accuracy			0.74	3577
macro avg	0.55	0.72	0.52	3577
weighted avg	0.94	0.74	0.81	3577

Accuracy Score: 0.7363712608331003
F1 Score: 0.2055602358887953

FIGURE 13. CLASSIFICATION REPORT OF LR TUNED MODEL

Berdasarkan *classification report* pada figure 13, dapat disimpulkan bahwa untuk kelas 0, ketepatannya adalah 0,98, yang menunjukkan bahwa dari semua contoh yang diprediksi berlabel 0,98% True Negative. Untuk kelas 1, ketepatannya adalah 0,12, menunjukkan bahwa hanya 12% dari contoh yang diprediksi sebagai kelas 1 adalah positif.

Untuk kelas 0, *recall* adalah 0,74, menunjukkan bahwa 74% dari True Negative diidentifikasi dengan benar sebagai kelas 0. Untuk kelas 1, *recall* adalah 0,71, menunjukkan bahwa 71% dari True Positive diidentifikasi dengan benar sebagai kelas 1.

Untuk kelas 0, skor F1 adalah 0,84, dengan mempertimbangkan keseimbangan antara *precision* dan *recall*. Untuk kelas 1, skor F1 adalah 0,21, yang mengindikasikan kinerja yang lebih rendah karena *precision* yang rendah dan *recall* yang sedang.

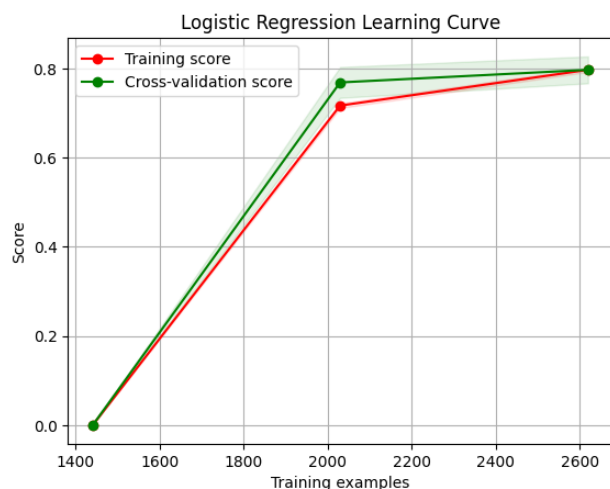
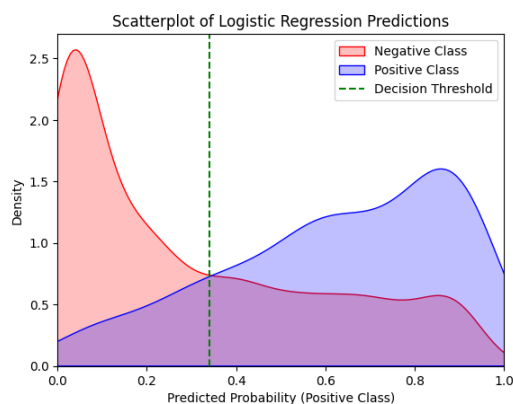


FIGURE 14. LR LEARNING CURVE

Kemudian, pada figure 14 merupakan hasil visualisasi *learning curve* dari model LR. Terlihat bahwa model LR memiliki performa yang baik karena model tidak mengalami *overfitting*, dilihat dari *recall score* yang dimiliki oleh LR yaitu sebesar 72%. Meskipun skor F1 dari model LR lebih kecil dibandingkan dengan model SVM



dan RF, skor *precision* dan *recall* milik model LR lebih tinggi dibandingkan dengan SVM ataupun RF, sehingga dapat disimpulkan bahwa algoritma LR memiliki potensi untuk *deployment*.

FIGURE 15. SCATTERPLOT MODEL LR

Selain itu, pada figure 15 merupakan hasil visualisasi dari *scatterplot* model *logistic regression* (LR) dalam melakukan prediksi. Dapat disimpulkan bahwa model LR masih sulit memprediksi pasien yang memiliki penyakit stroke dan yang tidak memiliki penyakit stroke dalam dataset test, sehingga model masih memiliki kekurangan yang perlu diperbaiki sebelum model dapat di *deploy*.

Terdapat *overlap* yang signifikan antara kelas positif dan negatif pada *scatterplot*, hal ini menunjukkan bahwa model mungkin masih mengalami kesulitan dalam keakuratan untuk membedakan kedua kelas tersebut berdasarkan probabilitas yang diprediksi.

Overlap menunjukkan bahwa mungkin ada kasus di mana probabilitas yang diprediksi untuk kelas positif sama atau bahkan lebih rendah daripada kelas negatif, yang mengarah ke potensi kesalahan klasifikasi.

Meskipun terdapat beberapa *overlap*, model ini masih dapat menangkap tren umum probabilitas yang lebih tinggi untuk kelas positif. Mungkin ada baiknya mempertimbangkan untuk menyesuaikan *decision threshold* untuk menemukan keseimbangan antara *precision* dan *recall*.

4) ARTIFICIAL NEURAL NETWORK (ANN)

Untuk melihat akurasi model, dibuat *classification report*. Terlihat pada figure 16 bahwa nilai akurasi yang didapatkan dari model ANN adalah sebesar 72.49% setelah melalui *hyperparameter tuning*.

```
112/112 [=====] - 0s 1ms/step - loss: 0.5141 - accuracy: 0.7249
Loss: 0.5141043663024902
Accuracy Score: 0.7249091267585754
```

	precision	recall	f1-score	support
0	0.98	0.73	0.83	3404
1	0.11	0.65	0.19	173
accuracy			0.72	3577
macro avg	0.54	0.69	0.51	3577
weighted avg	0.93	0.72	0.80	3577

FIGURE 16. CLASSIFICATION REPORT OF ANN TUNED MODEL

Berdasarkan *classification report* pada figure 16, dapat disimpulkan bahwa untuk kelas 0, ketepatannya adalah 0,98, yang menunjukkan bahwa dari semua contoh yang diprediksi berlabel 0,98% *True Negative*. Untuk kelas 1, ketepatannya adalah 0,11, menunjukkan bahwa hanya 11% dari contoh yang diprediksi sebagai kelas 1 adalah positif.

Untuk kelas 0, *recall* adalah 0,73, menunjukkan bahwa 73% dari *True Negative* diidentifikasi dengan benar sebagai kelas 0. Untuk kelas 1, *recall* adalah 0,65, menunjukkan bahwa 65% dari *True Positive* diidentifikasi dengan benar sebagai kelas 1.

Untuk kelas 0, skor F1 adalah 0,83, dengan mempertimbangkan keseimbangan antara *precision* dan *recall*. Untuk kelas 1, skor F1 adalah 0,19, yang mengindikasikan kinerja yang lebih rendah karena *precision* yang rendah dan *recall* yang sedang. Sementara itu, pada *figure 17* dan *figure 18* merupakan hasil visualisasi *training and validation loss*, serta *training and validation accuracy* dari model ANN.

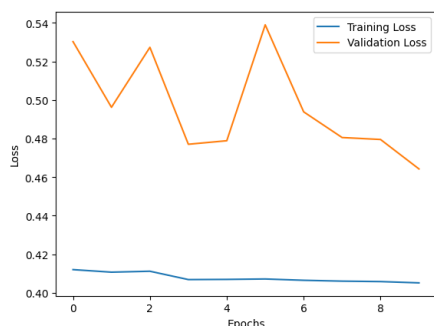


FIGURE 17. ANN MODEL TRAINING AND VALIDATION LOSS

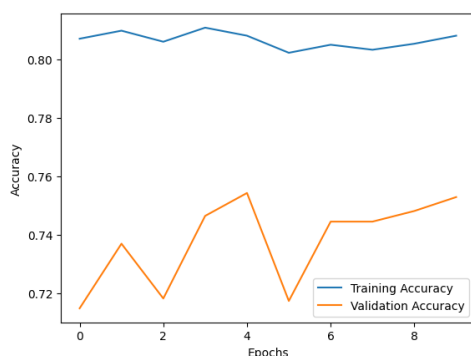


FIGURE 18. ANN MODEL TRAINING AND VALIDATION ACCURACY

Dari *figure 17*, *training loss* menurun secara konsisten dari waktu ke waktu, yang menunjukkan bahwa model belajar dan menyesuaikan diri dengan data pelatihan. *Validation loss* berfluktuasi tetapi tetap relatif stabil di seluruh epoch. Meskipun ada sedikit variasi, tren keseluruhan tidak meningkat secara signifikan, yang menunjukkan bahwa model tidak *overfitting*.

Di sisi lain pada *figure 18*, terlihat bahwa nilai *training accuracy* secara konsisten tinggi, menunjukkan bahwa model mampu menyesuaikan diri dengan baik dengan data pelatihan. Akurasi *validation* juga relatif tinggi dan tetap stabil, yang menunjukkan bahwa model dapat menggeneralisasi dengan baik untuk data yang belum pernah dilihat sebelumnya. Hal ini mendukung pengamatan bahwa model ini tidak *overfitting* dan memiliki performa yang cukup baik.

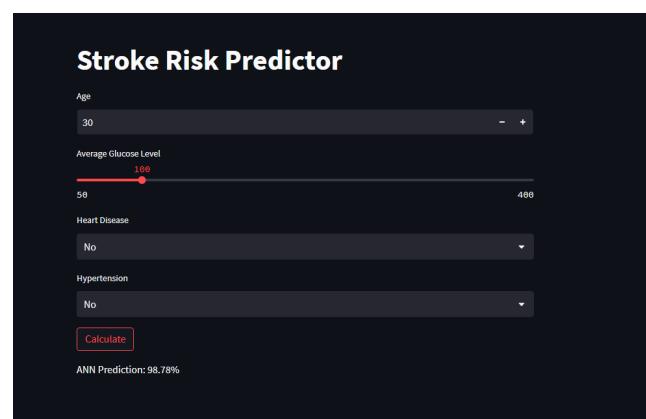
TABLE II
PERBANDINGAN AKURASI DAN F1 SCORE MODEL
SETELAH HYPERPARAMETER TUNING

Algoritma	Accuracy	F1-Score
Random Forest	86.07%	15.30%
SVM	69.89%	16.44%
Logistic Regression	73.63%	20.55%
Artificial Neural Network	72.49%	19%

Dengan hasil evaluasi model seperti pada *table II*, dipilih 2 model yang memiliki kemungkinan di *deploy* yaitu *logistic regression* dan ANN. Hal tersebut ditentukan atas dasar nilai skor f1, akurasi, dan *recall* kedua model yang baik.

F. MODEL DEPLOYMENT

Model *logistic regression* kemudian disimpan dalam *format .pkl*, dan ANN dalam *format .h5*, agar model dapat dipanggil kembali ketika telah siap di *deploy*.



FIGURE

V. CONCLUSION

Penelitian ini membuktikan bahwa melalui algoritma *machine learning* yang digunakan, prediksi Stroke mungkin dilakukan berdasarkan informasi yang diketahui dari tiap individu. Metode CRISP-DM (*Data Science*) sangat membantu dalam menganalisa data dan membuat proses penelitian menjadi lebih sederhana dan efisien tanpa perlu kehilangan arah dari tujuan bisnis, sehingga dapat mengambil keputusan yang tepat berdasarkan permasalahan bisnis yang dihadapi.

Keempat model algoritma (*random forest*, *SVM*, *logistic regression*, dan ANN) berhasil diimplementasikan dalam penelitian ini, namun hanya dua model yang dipilih sebagai model untuk *deployment* yaitu model *logistic regression* dan ANN karena performa nilai *precision*, *recall*, dan skor F1 kedua

model yang baik dan layak digunakan untuk deployment.

REFERENCES

- [1] F. S. Erlina, "10 Penyakit Penyebab Kematian Tertinggi di Indonesia (2019)," databoks, 7 February 2023. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2023/02/07/stroke-dan-tbc-masuk-dalam-10-penyakit-penyebab-kematian-tertinggi-di-indonesia>. [Accessed 11 June 2023].
- [2] B. Harmadi, "Penerapan algoritma decision tree pada prediksi risiko terserang stroke," *Doctoral dissertation, Universitas Islam Negeri Maulana Malik Ibrahim*, 2022.
- [3] M. N. Maskuri, Harliana, S. Kadek and M. H. B. Raden, "Penerapan Algoritma K-Nearest Neighbor (KNN) untuk Prediksi Penyakit Stroke," *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, vol. 4, no. 01, pp. 130-140, 2022.
- [4] A. V., "Klasifikasi Tingkat Risiko Penyakit Stroke Menggunakan Metode GA-Fuzzy Tsukamoto," *Universitas Brawijaya*, 2018.
- [5] C. I., F. F. and S. E., "Klasifikasi Tingkat Resiko Stroke," *Konferensi Nasional Sistem & Informasi*, pp. 11-13, 2016.
- [6] U. Amelia, I. Jamaludin and M. Anis, "Implementasi Algoritma Support Vector Machine (SVM) untuk Prediksi Penyakit Stroke dengan Atribut Berpengaruh," *Scientific Student Journal for Information, Technology and Science*, vol. 3, no. 2, pp. 254-259, 2022.
- [7] F. A. H. Atri, S. Tati and B. Agus, "KOMPARASI METODE KLASIFIKASI DATA MINING UNTUK PREDIKSI PENYAKIT STROKE," *E-Link: Jurnal Teknik Elektro dan Informatika*, vol. 18, no. 1, pp. 73-79, 2023.
- [8] S. Cheon, K. Jungyeon and L. Jihye, "The use of deep learning to predict stroke patient mortality," *International journal of environmental research and public health*, vol. 16, no. 11, p. 1876, 2019.
- [9] K. R. Sulaeman, S. Cati and E. S. Randy, "Analisis Algoritma Support Vector Machine Dalam Klasifikasi Penyakit Stroke," *Proceedings of Engineering*, vol. 9, no. 3, 2022.
- [10] F. Handayani, "Komparasi Support Vector Machine, Logistic Regression Dan Artificial Neural Network Dalam Prediksi Penyakit Jantung," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 7, no. 3, pp. 329-334, 2021.
- [11] W. Hidayat, A. Mursyid and S. Arief, "Pengaruh Algoritma ADASYN dan SMOTE terhadap Performa Support Vector Machine pada Ketidakseimbangan Dataset Airbnb," *Edumatic: Jurnal Pendidikan Informatika*, vol. 5, no. 1, pp. 11-20, 2021.
- [12] E. Sutoyo and A. F. M., "Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Television Advertisement Performance Rating Menggunakan Artificial Neural Network," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 6, no. 3, pp. 379-385, 2020.
- [13] G. Sailasya and L. A. K. Gorli, "Analyzing the performance of stroke prediction using ML classification algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021.
- [14] A. T. R. Jose, "Stroke prediction through Data Science and Machine Learning Algorithms," 2021.
- [15] S. Huber, W. Hajo, S. Dorothea and I. Steffen, "DMME: Data mining methodology for engineering applications—a holistic extension to the CRISP-DM model," *Procedia Cirp*, vol. 79, pp. 403-408, 2019.
- [16] J. L. Speiser, E. M. Michael, T. Janet and I. Edward, "A comparison of random forest variable selection methods for classification prediction modeling," *Expert systems with applications*, vol. 134, pp. 93-101, 2019.
- [17] I. Goodfellow, B. Yoshua and C. Aaron, *Deep learning*, MIT Press, 2016.
- [18] D. Berrar, "Cross-Validation," pp. 542-545, 2019.
- [19] Y. Sasaki, "The truth of F-measure," 26 October 2007. [Online]. Available: <https://www.cs.odu.edu/~mukka/cs795sum11dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf>. [Accessed 22 March 2023].
- [20] J. Korstanje, "The F1 score," Medium, 31 August 2021. [Online]. Available: <https://towardsdatascience.com/the-f1-score-bec2bbc38aa6>. [Accessed 22 March 2023].



FIONA TANADI was born in Jakarta in 2001. She is currently pursuing computer science bachelor degree, with Big Data Analytics as the field of study in Universitas Multimedia Nusantara.

Her research interests include machine learning, business intelligence and big data analytics.



ABDUL RAZAK DANISH was born in Brunei Darussalam in 2000.

he is currently pursuing education in Universitas Multimedia Nusantara going for the major of Information Systems. His interest include Big Data analytic ,machine learning and Enterprise Resource Planning .



LEONARD ALDO was born in Jakarta in 1999, currently pursuing education in Multimedia Nusantara University going for the major of Information Systems. His Interests includes Machine learning, Deep Learning, and Big Data Analytics