



## Deep Neural Network untuk Prediksi Stroke

Anas Faisal<sup>#1</sup>, Agus Subekti<sup>#2</sup>

<sup>#</sup>Program Studi Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Nusa Mandiri  
Jalan Kramat Raya No. 18, Senen, Jakarta Pusat

<sup>1</sup>14002356@nusamandiri.ac.id

<sup>2</sup>agus@nusamandiri.ac.id

**Abstrak**— Pada Tahun 2019 Organisasi Kesehatan Dunia (WHO) mendudukan stroke sebagai tujuh dari sepuluh penyebab utama kematian. Kementerian Kesehatan menggolongkan stroke sebagai penyakit katastropik karena dampaknya luas secara ekonomi dan sosial. Oleh karena itu, diperlukan peran dari teknologi informasi untuk memprediksi stroke guna pencegahan dan perawatan dini. Analisis data yang memiliki kelas tidak seimbang mengakibatkan ketidakakuratan dalam memprediksi stroke. Penelitian ini membandingkan tiga teknik oversampling untuk mendapatkan model prediksi yang lebih baik. Data kelas yang sudah diseimbangkan diuji menggunakan tiga model Arsitektur Deep Neural Network (DNN) dengan melakukan optimasi pada beberapa parameter yaitu optimizer, learning rate dan epoch. Hasil paling baik didapatkan teknik oversampling SMOTETomek dan Arsitektur DNN dengan lima hidden layer, optimasi Adam, learning rate 0.001 dan jumlah epoch 500. Skor akurasi, presisi, recall, dan f1-score masing-masing mendapatkan 0.96, 0.9614, 0.9608 dan 0.9611.

**Kata kunci**— Deep Learning, Deep Neural Network, SMOTE, Tomek links, Stroke

### I. PENDAHULUAN

Stroke (*Cerebro-Vascular Accident* atau CVA) merupakan sebuah kondisi ketika gangguan mendadak dalam aliran darah ke otak, yang disebabkan oleh penyumbatan (stroke iskemik) atau semburan (hemoragik stroke) pembuluh darah yang membawa oksigen dan nutrisi[1]. Faktor risiko umum termasuk jangka panjang antara lain hipertensi, hyperglikemia, hyperlipidemia dan tinggi tekanan, stres emosional, dapat menyebabkan stroke. Hal ini diyakini, di bawah asumsi kedokteran, pencegahan deteksi faktor risiko ini sangat membantu untuk perawatan dini [2]. Penyediaan model yang dapat memprediksi risiko stroke akan sangat berkontribusi secara signifikan terhadap pencegahan dan perawatan dini[3]. Namun, volume data yang tinggi, heterogenitas dan kompleksitas yang terdapat pada data medis menjadi tantangan terbesar dalam memprediksi penyakit stroke.

Prediksi stroke pada pasien bertujuan untuk mengurangi potensi kematian yang disebabkan stroke. Model prediksi dengan, pembelajaran mesin telah diusulkan, antara lain

menggunakan *Chi-Square (Chi-2)*, *Decision Tree* [3], *Two-Class Boosted Decision Tree* [4], *Naive Bayes*, *Support Vector Machine* [5], *Logistic Regression*, *Random Forest*, *Gradient Boosting* [6]. Metoda yang diusulkan pada referensi tersebut diuji dengan menggunakan dataset yang berbeda-beda dan menghasilkan nilai akurasi yang bervariasi. Hal ini menjadi kurang tepat untuk membandingkan antar hasil dari berbagai algoritma *machine learning* tersebut. Hasil dari pengujian secara *distributed environment* didapatkan nilai tertinggi menggunakan algoritma *Gradient Boosting* dengan akurasi sebesar 94,49% [6].

Selain itu, C. C. Peng, et al melakukan penelitian dengan preprocessing *Feature Elimination with Cross Validation*, dilakukan proses pada kondisi 1000 kali *cross validation* mendapatkan nilai akurasi 98% [2]. S. Cheon and J. Kim menggunakan metode *Principal Component Analysis* dan pendekatan *Deep Neural Network (DNN)* [7]. Sedangkan A. Fitri, N. Masruriah, T. Djatna et al juga menggunakan *Artificial Neural Network* untuk melakukan prediksi stroke, dengan nilai Akurasi terbaik sebesar 94,97% [8].

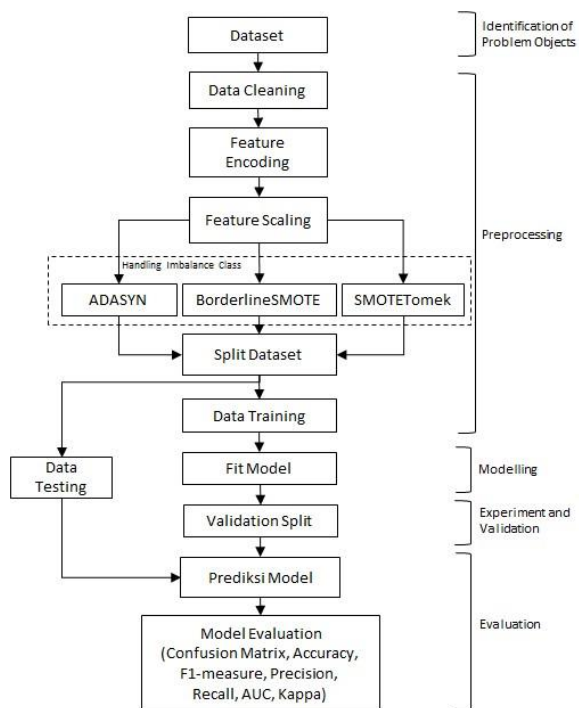
Salah satu metode yang prospek untuk menghasilkan akurasi yang lebih tinggi dibandingkan dengan algoritma *Gradient Boosting* [9] adalah menggunakan *Deep Learning*. Pada *Deep Learning* terdapat sebuah Arsitektur dan *hyperparameter* yang harus dilakukan *setup* untuk mendapatkan nilai maksimal. Kelemahan pada [6] tidak menggunakan variasi *SMOTE* untuk *oversampling*. Berbagai variasi teknik *SMOTE* dapat dipilih untuk meningkatkan nilai akurasi. Pada makalah ini, kami mengusulkan metode deep learning dan melakukan *preprocessing* pada dataset *training*. Pada *pre-processing* dilakukan imputasi untuk menangani *missing values* dan teknik *SMOTETomek* untuk menangani *imbalance class*. Pemodelan arsitektur DNN dilakukan dengan menggunakan *library* Keras dan *Back-end Tensorflow*. Arsitektur dengan *hyperparameter* yang kami usulkan mendapatkan nilai akurasi sebesar 96%, precision 96.14%, recall 96.08%, dan f-measure sebesar 96,11%.

Makalah ini disusun sebagai berikut: Bagian II menjelaskan metode yang kami usulkan. Bagian ini membahas eksplorasi dataset yang digunakan sampai dengan arsitektur model DNN. Bagian III menjelaskan

proses eksperimen dan Bagian IV membahas dan menganalisis hasil penelitian. Akhirnya, pada Bagian V, kami membuat kesimpulan hasil penelitian.

II. METODOLOGI PENELITIAN

Tahapan-tahapan penelitian dilakukan sebagaimana disajikan pada Gambar 1. Tahapan tersebut dimulai dari proses pengambilan data dari *Kaggle* sampai dengan melakukan evaluasi model prediksi menggunakan data testing.



Gambar 1. Kerangka pemikiran metode penelitian

A. Dataset Description

Penelitian menggunakan dataset *stroke prediction* yang diunduh dari [10]. Penggunaan data tersebut agar dapat diperbandingkan dengan metode yang sudah dilakukan oleh penelitian sebelumnya, antara lain [6]. Disamping itu, penelitian dilakukan untuk mengatasi kelemahan metode yang dilakukan peneliti sebelumnya. Fokus penelitian pada

kebaruan metode, sehingga akan lebih baik dengan menggunakan dataset yang sama. Data tersebut terdiri atas 5110 instan dengan 12 Atribut. Kedua belas atribut ini dapat dibagi menjadi tiga besar yaitu: atribut yang tidak dapat dikontrol, atribut yang merupakan gaya hidup dan atribut faktor risiko medis. Atribut yang tidak dapat dikontrol merupakan atribut yang melekat secara otomatis pada pasien termasuk jenis kelamin dan usia. Atribut gaya hidup merupakan kebiasaan setiap individu yang biasanya berdasarkan keinginan dan kemampuan ekonomi. Sedangkan atribut faktor risiko medis merupakan variabel-variabel yang berhubungan dengan peningkatan risiko suatu penyakit atau infeksi tertentu.

Atribut dan penjelasannya disajikan pada Tabel 1. Berdasarkan penjelasan tersebut, dataset memiliki enam atribut *binary*, delapan atribut dalam bentuk kategorikal dan tiga atribut dalam bentuk numerik. Sedangkan untuk informasi terkait dengan statistik deskriptif disajikan pada Tabel 2.

TABEL I  
DESKRIPSI DATASET

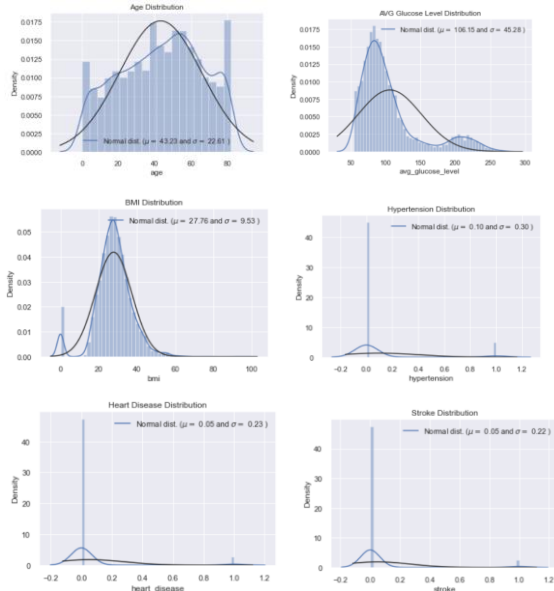
Atribut	Description
ID	Nomor id pasien
Gender	Jenis kelamin pasien
Age	Usia pasien
Hypertension	0 – tidak hipertensi 1 – hipertensi
Heart disease	0 – tidak memiliki riwayat penyakit jantung 1 – memiliki riwayat penyakit jantung
Marital status	Menikah atau tidak menikah
Work-type	Jenis pekerjaan pasien
Residence area	Wilayah tempat tinggal pasien
Avg-glukose	Rata-rata level glukosa dalam darah yang diukur setelah makan
BMI	Body Mass Index pasien
Smoking status	Status merokok pasien
Stroke status	0 – tidak stroke 1 – stroke

Pada Gambar 2, kita dapat mengamati bahwa untuk usia, rata-rata level glukosa dan indeks masa tubuh sebagai faktor yang harus dipertimbangkan dalam memprediksi kemungkinan penyakit stroke. Pada feature usia berbentuk

TABEL II  
DESCRIPTIVE STATISTIC

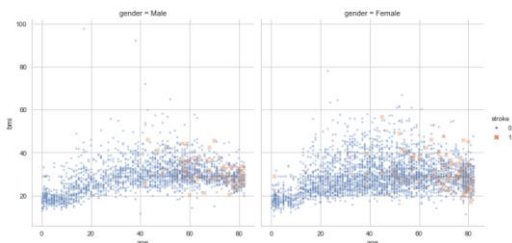
	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	5110.00	5110.00	5110.00	5110.00	5110.00	4909.00	5110.00
mean	36517.83	43.23	0.09	0.05	106.14	28.89	0.048
std	21161.72	22.61	0.29	0.22	45.28	7.85	0.21
min	67.00	0.08	0.00	0.00	55.12	10.30	0.00
25%	17741.25	25.00	0.00	0.00	77.24	23.50	0.00
50%	36932.00	45.00	0.00	0.00	91.88	28.10	0.00
75%	54682.00	61.00	0.00	0.00	114.09	33.10	0.00
max	72940.00	82.00	1.00	1.00	271.74	97.60	1.00

unimodal yang mengindikasikan bahwa rata-rata usia yang paling banyak mengalami stroke pada usia 43 tahun dengan standar deviasi 22.61 dan hampir rata di semua usia kecuali adanya lonjakan yang tajam pada usia anak dan orang tua. Rata-rata level glukosa berbentuk bimodal dengan sedikit *skew-right*. Nilai standar deviasi rata-rata level glukosa sebesar 45 dan nilai rata-rata level glukosa sebesar 106. Indeks massa tubuh lebih berbentuk unimodal dan sedikit miring ke kiri (*skew-left*) dengan nilai standar deviasi sebesar 7 dan nilai rata-rata sebesar 128.

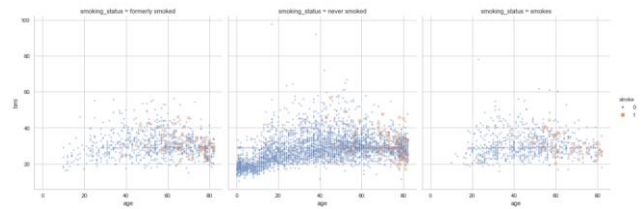


Gambar 2. Distribusi Kepadatan *features*

Pada Gambar 3 kita dapat mengamati pengaruh usia, indeks massa tubuh dan jenis kelamin terhadap stroke. Usia diatas 40 tahun lebih dominan terkena stroke baik pada jenis kelamin laki-laki maupun perempuan. Sedangkan pada Gambar 4 kita dapat menganalisis bahwa status merokok juga menjadi faktor yang sangat mempengaruhi terhadap stroke. Sebagian besar yang tidak stroke adalah yang tidak pernah merokok terutama pada usia dibawah 40 tahun.



Gambar 3. Pengaruh indeks massa tubuh, usia dan jenis kelamin pada stroke



Gambar 4. Pengaruh Indeks Massa Tubuh, Usia dan Status Merokok pada Stroke

**B. Missing Values**

Berdasarkan observasi, dataset memiliki *missing values* pada tiga atribut, yaitu *gender*, indeks massa tubuh dan *smoking status*. Pada *gender* hanya terdapat satu instan yang tidak diketahui klasifikasinya dan langsung dilakukan imputasi dengan teknik *modus*. *Missing values* pada indeks massa tubuh sebanyak 201 instan (3,9%) dan *missing values* untuk *smoking status* sebanyak 1544 (30,2%).

**C. Imbalance Class**

Dataset memiliki dua target *class* dengan komposisi yang tidak seimbang sebagaimana disajikan pada Gambar 5. Target *class* 0 (tidak-stroke) sebanyak 4861 *instance* (95,13%) dan *class* 1 (stroke) sebanyak 249 *instance* (4,87%).

**D. Synthetic minority oversampling technique (SMOTE)**

Metode *SMOTE* diusulkan oleh Chawla et al. (2002) [11] dengan menghasilkan pengamatan sintetik untuk kelas minoritas. Untuk observasi kelas minoritas, observasi sintetik dihasilkan dalam rentang acak antara pengamatan dan *k-nearest neighbor* kelas minoritas. Prosedur ini dilakukan untuk setiap observasi kelas minoritas. Untuk *SMOTE*, jumlah *k-nearest neighbor* diatur ke angka 5. Meskipun *SMOTE* cukup efektif untuk meningkatkan akurasi klasifikasi data minoritas, tetapi masih ada masalah, antara lain terjadinya overgeneralisasi. Data sintesis yang dihasilkan oleh *SMOTE* masih dimungkinkan untuk menyebar pada data minoritas dan mayoritas, sehingga akan mengurangi kinerja klasifikasi. Rumus untuk menghasilkan data sintesis oleh *SMOTE* adalah sebagai berikut[12]:

$$x^{synthetic} = x^i + (x^j - x^i) \times \delta \dots\dots\dots (1)$$

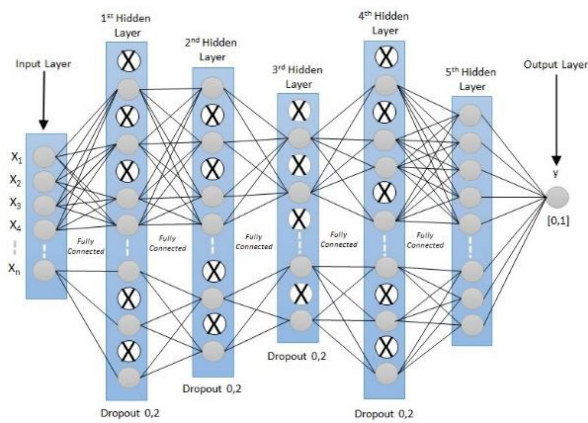
Dimana  $x^{synthetic}$  adalah data sintesis,  $x^i$  adalah instan kelas minoritas,  $x^j$  adalah instan yang dipilih secara acak dari *K-nearest neighbor* dari  $x^i$  instan minoritas, dan  $\delta$  adalah vektor di mana setiap elemen adalah angka acak dari [0,1].

Pada makalah ini kami melakukan eksperimen dengan menyeimbangkan kelas dengan tiga variasi teknik *SMOTE*, yaitu: *Adaptive Synthetic (Adasyn)*, *BorderlineSMOTE*, dan *SMOTETomek links*.

**E. Architecture Deep Neural Network**

Pada makalah ini, kami mengusulkan arsitektur *DNN* dengan algoritma *backpropagation*. Model arsitektur *DNN*

yang kami usulkan memiliki lima *hidden layer*. Pada setiap layer, kami melakukan *setup hyperparameter* untuk jumlah *neuron*, *activation function*, *optimizer* dan *regularization*. Pada *hidden layer* masing-masing memiliki *neuron* sebanyak 320, 288, 32, 320 dan 32 units. *Activation function* pada *input layer* dan setiap *hidden layer* menggunakan *Rectified linear activation function (ReLU)* [13], sedangkan untuk *output layer* menggunakan *Sigmoid activation function* [14]. *Optimizer* yang digunakan *Adam* [15] dengan *learning rate* sebesar 0,001. Untuk mengurangi *overfitting*, kami menggunakan teknik regularisasi dengan melakukan *drop out* pada *hidden layer* sebesar 0,2[16]. Jumlah *epoch* yang digunakan sebesar 500 dengan ukuran *batch* sebesar 96. Gambar 5 menunjukkan arsitektur DNN yang kami usulkan.



Gambar 5. Arsitektur Deep Neural Network

III. EKSPERIMEN

Pada bagian ini kami melakukan eksperimen dengan tahapan: 1) *Pre-processing*, 2) Model Eksperimen.

A. *Pre-processing*

Dataset yang digunakan memiliki karakteristik *missing values*, terdapat nilai *categorical text*, memiliki skala yang tidak standar dan *high imbalance class*. *Missing values* ditangani dengan *mean imputation* untuk *feature* status merokok. *Feature* yang memiliki nilai *categorical text* dilakukan *feature encode* dengan *label encoding*. *Standardscaler* digunakan untuk standarisasi data [17]. Teknik *SMOTE* digunakan untuk menangani ketidakseimbangan kelas[11]. Eksperimen dilakukan pada setiap model dengan menggunakan tiga teknik *SMOTE*, yaitu: *Adasyn*, *BorderlineSMOTE*, dan *SMOTETomek*. Teknik *SMOTE* diimplementasikan menggunakan *imbalance-learn libraries*.

Dataset dibagi menjadi dua bagian yaitu 80% sebagai data training dan 20% sebagai data testing. Proses pembagian tersebut dilakukan secara random. Pembagian dilakukan dengan menggunakan *scikit-learn libraries* [18]. Selanjutnya data training dibagi menjadi dua bagian yaitu 80% sebagai data training dan 20% bagian akhir sebagai *data validation*.

B. Model Eksperimen

Eksperimen dilakukan dengan membuat tiga model arsitektur DNN yang terdiri dari 4, 5, dan 6 *hidden layer* sebagaimana disajikan pada Tabel 3. Setiap model memiliki jumlah *neuron* yang bervariasi. Ketiga model kemudian digunakan untuk memproses dataset yang telah diseimbangkan dengan ketiga variasi *SMOTE*. Parameter Model DNN untuk fungsi aktivasi *input layer* dan *hidden layer* menggunakan *ReLU* dan *output layer* menggunakan *sigmoid*, jumlah *epoch* sebesar 400, 500 dan 600 sebagaimana disajikan pada Tabel 4. Pada proses training dioptimasi dengan *Stochastic Gradient Descent (SGD)*, *Adaptive moment estimation (Adam)*, dan *Root Mean Squared propagation (RMSprop)* dan variasi nilai *learning rate* sebesar 0.1, 0.01, 0.001 dan 0.0001 sebagaimana disajikan pada Tabel 5.

TABEL III ARSITEKTUR DNN

Model DNN		Jumlah		
Mdl	Teknik SMOTE	Input Neuron	Hidden n Layer	Hidden Neuron
A	Adasyn	10	4	320,256,32,96
	BLSMOTE	10		320,256,32,64
	SMOTETomek	10		320,256,32,32
B	Adasyn	10	5	320,288,32,320,96
	BLSMOTE	10		320,288,32,320,64
	SMOTETomek	10		320,288,32,320,32
C	Adasyn	10	6	320,224,128,32,320,96
	BLSMOTE	10		320,224,128,32,320,64
	SMOTETomek	10		320,224,128,32,320,32

TABEL IV PARAMETER MODEL DNN

Parameter	Nilai Parameter
Aktivasi Input Layer	ReLU (Rectified Linear Unit)
Aktivasi Hidden Layer	ReLU (Rectified Linear Unit)
Aktivasi Output	Sigmoid
Dropout	0,2
Epoch	400,500,600
Batch Size	96

TABEL V PARAMETER OPTIMASI MODEL DNN

Optimasi	Learning Rate	Momentum
SGD (Stochastic Gradient Descent)	0.1	0.9
	0.01	0.9
	0.001	0.9
Adam (Adaptive Moments Estimation)	0,01	Default
	0,001	Default
	0,0001	Default
RMSprop (Root Mean Square propagation)	0,01	Default
	0,001	Default
	0,0001	Default



IV. HASIL DAN PEMBAHASAN

Dataset yang digunakan memiliki dua kelas (*binary classification*) yaitu: 1 (stroke) dan 0 (non-stroke). Model DNN dievaluasi dengan menggunakan standard *metrics performance* [19]. *Metrics performance* tersebut sebagai berikut: *Accuracy*, *Precision*, *Recall*, *f1-score*, dan *ROC-AUC*.

*Accuracy* didefinisikan sebagai rasio antara sampel yang diklasifikasikan dengan benar ke jumlah total Sampel [20]. *Accuracy* dapat dihitung dengan menggunakan rumus sebagai berikut[21]:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \dots\dots\dots (2)$$

*Precision* adalah rasio sampel berkategori positif yang diklasifikasi dengan benar dibandingkan dengan total sampel yang diklasifikasi sebagai positif.

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots (3)$$

*Recall* adalah rasio sampel yang diprediksi positif yang diidentifikasi dengan benar.

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots (4)$$

*F1-Score* merupakan rata-rata harmonik antara *precision* dan *recall*.

$$F1 - score = \frac{2x (Precision \times Recall)}{(Precision+Recall)} \dots\dots\dots (5)$$

*ROC-AUC* merupakan grafik hubungan dua dimensi antara *True Positive Rate* (*y-axis*) dengan *False Positive Rate* (*x-axis*). Metode penilaian grafis seperti *ROC* dan kurva *Precision-Recall* memberikan interpretasi yang berbeda dari kinerja klasifikasi.

Agar dapat dibandingkan dengan penelitian sebelumnya, maka salah satu *metric* yang digunakan sebagai dasar pemilihan model usulan prediktif yaitu *accuracy score*. Berdasarkan Tabel 6, *accuracy score* paling baik didapatkan model B dengan optimasi *Adam*, *learning rate* 0.001, jumlah *epoch* 500 dan teknik *SMOTE SMOTETomek*. Dataset yang digunakan merupakan dataset yang imbalance, *metric accuracy* kurang relevan karena akan menimbulkan bias. Kesalahan memprediksi kelas minoritas (*False Positive-precision*) biayanya lebih tinggi daripada kelas mayoritas (*False Negative-recall*), sehingga memprediksi kelas minoritas dengan benar (*True Positive*) lebih penting[19]. Namun, memilih model berdasarkan hal tersebut harus berani mengambil risiko yang diakibatkan dari tingkat kesalahan klasifikasi yang mungkin terjadi oleh masing-masing *metric*. Oleh karena itu, untuk pemilihan model selain menggunakan *accuracy score*, juga menggunakan kombinasi antara *precision* dan *recall*, yaitu menggunakan *metric F1-score* atau *f-measure* yang merupakan nilai rata-rata harmonik antara *precision* dan *recall*.

Berdasarkan Tabel 7, *precision* paling baik didapatkan oleh Model C dengan optimasi *RMSprop*, *learning rate* 0.01, jumlah *epoch* 400 dan teknik *SMOTE Adasyn*. Sedangkan berdasarkan Tabel 8, *recall* paling baik juga didapatkan oleh Model C dengan optimasi *SGD* dan *Adam* dengan masing-masing *learning rate* 0.001 dan 0.0001, jumlah *epoch* 400 dan teknik *SMOTE Adasyn*. Hal ini menunjukkan bahwa untuk *precision* yang nilainya paling tinggi, *recall* nya lebih rendah dibandingkan Model lainnya. Begitu juga sebaliknya untuk Model yang *recall*nya paling tinggi mendapatkan *precision* lebih rendah dibandingkan Model yang lainnya. Hal ini yang dinamakan *trade-off precision recall* [22].

TABEL VI  
PERBANDINGAN ACCURACY SCORE

Optimasi	SMOTE	Model	Epoch	LR	Acc
SGD	Adasyn	B	500	0,1	0,9493
	Borderline SMOTE	C	400	0,1	0,9527
	SMOTE Tomek	A	600	0,1	0,9496
Adam	Adasyn	C	400	0,001	0,9508
	Borderline SMOTE	B	400	0,01	0,9573
	SMOTE Tomek	B	500	0,001	0,9600
RMSprop	Adasyn	C	500	0,01	0,9549
	Borderline SMOTE	C	400	0,01	0,9553
	SMOTE Tomek	B	600	0,01	0,9563

TABEL VII  
PERBANDINGAN PRECISION

Optimasi	SMOTE	Mdl	Epch	LR	Prec.
SGD	Adasyn	B	500	0,1	0,9131
	Borderline SMOTE	C	400	0,1	0,9279
	SMOTE Tomek	C	500	0,1	0,9175
Adam	Adasyn	A	400	0,01	0,9208
	Borderline SMOTE	A	600	0,01	0,9368
	SMOTE Tomek	A	400	0,01	0,9296
RMSprop	Adasyn	C	400	0,01	0,9391
	Borderline SMOTE	B	400	0,001	0,9345
	SMOTE Tomek	B	500	0,01	0,9337

TABEL VIII  
PERBANDINGAN RECALL

Optimasi	SMOTE	Mdl	Epch	LR	Recall
SGD	Adasyn	C	400	0,001	0,9969
	Borderline SMOTE	C	500	0,001	0,9845

Optimasi	SMOTE	Mdl	Epch	LR	Recall
	SMOTE Tomek	C	600	0,01	0,9947
Adam	Adasyn	C	400	0,0001	0,9969
	Borderline SMOTE	C	400	0,001	0,9845
	SMOTE Tomek	A	400	0,001	0,9957
RMSprop	Adasyn	B	500	0,0001	0,9948
	Borderline SMOTE	B	400	0,0001	0,9835
	SMOTE Tomek	B	400	0,01	0,9936

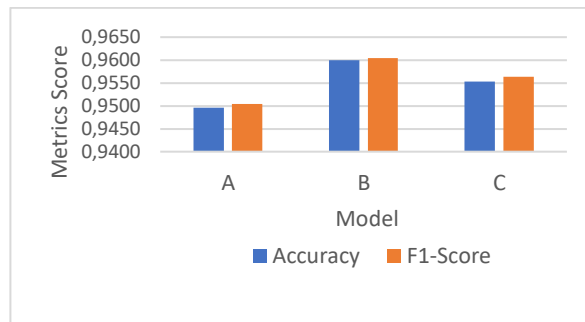
Berdasarkan Tabel 6 dan Tabel 9, kita dapat menganalisis bahwa hasil *accuracy score* dan *f1-score* paling baik didapatkan oleh Model B dengan jumlah *epoch* 500, optimasi Adam, *learning rate* 0.001 dan teknik SMOTETomek. Metode gabungan menggunakan SMOTE sebagai teknik pengambilan sampel data dengan menggabungkan Tomek links sebagai teknik pembersihan data merupakan algoritma *pre-processing* yang sangat kuat untuk mengatasi masalah kelas yang tidak seimbang. Dengan menggabungkan SMOTE dengan teknik Tomek links, batas antara yang berbeda kelas dapat dengan mudah diidentifikasi[23]. Sehingga menghasilkan tingkat akurasi lebih baik. Perbandingan *accuracy* dan *f1-score* juga ditunjukkan pada gambar 6, bahwa model B mendapatkan *accuracy score* dan *f1-score* yang lebih baik dibandingkan kedua model lainnya. Sehingga model yang dipilih sebagai model prediktif adalah Model B.

TABEL IX  
PERBANDINGAN F1-SCORE

Optimasi	SMOTE	Model	Epoch	LR	F1-Score
SGD	Adasyn	B	500	0,1	0,9503
	Borderline SMOTE	C	400	0,1	0,9539
	SMOTE Tomek	A	600	0,1	0,9504
Adam	Adasyn	C	400	0,001	0,9516
	Borderline SMOTE	B	400	0,01	0,9582
	SMOTE Tomek	B	500	0,001	0,9604
RMSprop	Adasyn	C	500	0,01	0,9551
	Borderline SMOTE	C	400	0,01	0,9563
	SMOTE Tomek	B	600	0,01	0,9565

Model yang telah dipilih sebagai model prediktif kemudian dievaluasi menggunakan data *testing* dan dibandingkan hasilnya dengan penelitian sebelumnya. Perbandingan *metrics performance* dengan penelitian sebelumnya sebagaimana disajikan pada Tabel 10 dan Gambar 7. Berdasarkan tabel tersebut dapat dianalisa bahwa skor *accuracy*, *precision*, *recall*, *f1-score* dan *AUC*

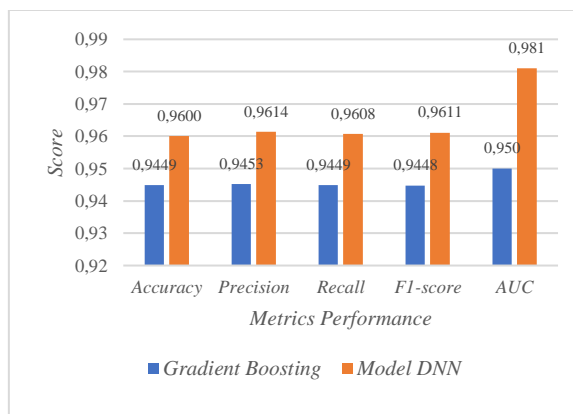
yang didapatkan model DNN lebih besar daripada menggunakan model Gradient Boosting.



Gambar 6. Perbandingan Metric Accuracy dan f1-score antar Model

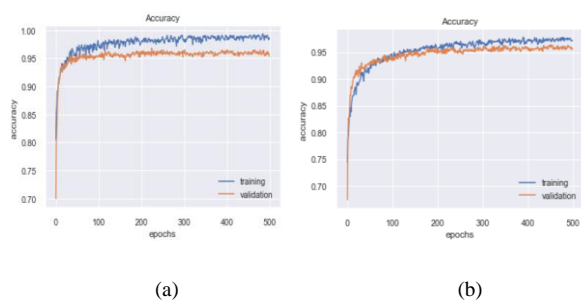
TABEL X  
PERBANDINGAN METRIC PERFORMANCE MODEL DNN DENGAN GRADIENT BOOSTING

Model	Acc.	Prec	Recall	F1-score	AUC
Gradient Boosting	0,9449	0,9453	0,9449	0,9448	0,950
Model DNN	0,9600	0,9614	0,9608	0,9611	0,981

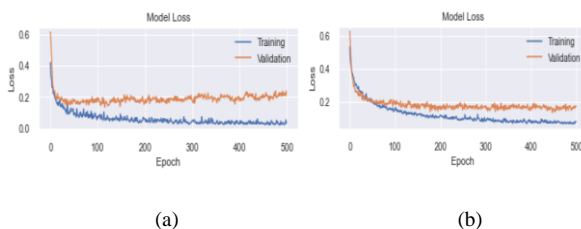


Gambar 7. Perbandingan Metric Performance Model DNN dan Gradient Boosting

Eksperimen juga memastikan bahwa model yang telah dipilih tidak *over-fitting*, maka dilakukan analisis terhadap grafik *history training* dan *validation accuracy* dan *logarithmic loss*. Gambar 8 (a) menunjukkan pergerakan *history training* dan *validation accuracy* sebelum diberikan nilai *dropout*. Pada Grafik tersebut menunjukkan *accuracy score* untuk *training* yang terus meningkat, namun ketika dilakukan validasi selama proses *training accuracy score*-nya bergerak turun. Hal ini akan mengakibatkan model lemah dalam memprediksi data diluar pola *dataset training*. Gambar 8 (b) menunjukkan pergerakan *history training* dan *validation accuracy* setelah diberikan nilai *dropout*. Grafik *accuracy training* bergerak naik dan disertai dengan naiknya *accuracy validation*, hal ini mengindikasikan bahwa model tidak *over-fitting*. Analisis sebaliknya dilakukan pada *history logarithmic loss* yang disajikan pada Gambar 9 (a) dan (b).



Gambar 8. Perbandingan *training* dan *validation accuracy history* sebelum dan sesudah *dropout*



Gambar 9. Perbandingan *training* dan *validation loss history* sebelum dan sesudah *dropout*

### V. KESIMPULAN

Penelitian untuk mendapatkan model prediksi stroke yang lebih baik telah dilakukan. Berdasarkan hasil model prediksi tersebut dijadikan sebagai salah satu informasi untuk memberikan saran kemungkinan potensi stroke. Dataset medis seringkali memiliki kelas yang tidak seimbang. Model arsitektur DNN yang kami usulkan dengan *pre-processing SMOTE* dan dikombinasikan dengan *Tomek links* menghasilkan *performance accuracy*, *f1-score* dan *AUC* sebesar 0.96, 0.9611 dan 0.981. *Performance* tersebut lebih tinggi dibandingkan *performance model* yang dihasilkan penelitian sebelumnya yang menggunakan klasifikasi *Extreme Gradient Boosting*.

Penelitian selanjutnya akan dilakukan dengan dataset dengan dimensi yang lebih besar, karena faktor-faktor medis sangat kompleks dan dilakukan pemodelan dengan menggunakan model *deep learning* lainnya.

### REFERENSI

[1] A. Fitri, N. Masruriyah, T. Djatna, M. K. D. Hardhienata, H. H. Handayani, and D. Wahiddin, "Predictive Analytics for Stroke Disease," pp. 13–16, 2020, doi: 10.1109/ICIC47613.2019.8985716.

[2] C. C. Peng, S. H. Wang, S. J. Liu, Y. K. Yang, and B. H. Liao, "Artificial Neural Network Application to the Stroke Prediction," *2nd IEEE Eurasia Conf. Biomed. Eng. Healthc. Sustain. 2020, ECBIOS 2020*, pp. 130–133, 2020, doi: 10.1109/ECBIOS50299.2020.9203638.

[3] P. Songram and C. Jareanpon, "A Study of Features Affecting on Stroke Prediction Using Machine Learning," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11909 LNAI, pp. 216–225, 2019, doi: 10.1007/978-3-030-33709-4\_19.

[4] S. Ray, K. Alshouli, A. Roy, A. Alghamdi, and D. P. Agrawal, "Chi-Squared Based Feature Selection for Stroke Prediction using AzureML," *2020 Internat. Eng. Technol. Comput. IETC 2020, 2020*, doi: 10.1109/IETC47856.2020.9249117.

[5] P. Chantamit-O-Pas and M. Goyal, "Prediction of stroke using deep learning model," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10638 LNCS, pp. 774–781, 2017, doi: 10.1007/978-3-319-70139-4\_78.

[6] M. Rajora, M. Rathod, and N. S. Naik, "Stroke Prediction Using Machine Learning in a Distributed Environment," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12582 LNCS, pp. 238–252, 2021, doi: 10.1007/978-3-030-65621-8\_15.

[7] S. Cheon and J. Kim, "The Use of Deep Learning to Predict Stroke Patient Mortality," 2019, doi: doi.org/10.3390/ijerph16111876.

[8] A. F. N. Masruriyah, T. Djatna, M. K. D. Hardhienata, H. H. Handayani, and D. Wahiddin, "Predictive Analytics For Stroke Disease," pp. 13–16, 2020, doi: 10.1109/ICIC47613.2019.8985716.

[9] M. Rajora, M. Rathod, and N. S. Naik, "Stroke prediction using machine learning in a distributed environment," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12582 LNCS, pp. 238–252, 2021, doi: 10.1007/978-3-030-65621-8\_15.

[10] Fedesoriano, "Stroke Prediction Dataset," *Kaggle*, 2021. <https://www.kaggle.com/fedesoriano/stroke-prediction-dataset> (accessed Mar. 06, 2021).

[11] B. Kovács, F. Tinya, C. Németh, and P. Ódor, "SMOTE: Synthetic Minority Over-sampling Technique," *Ecol. Appl.*, vol. 30, no. 2, pp. 321–357, 2020, doi: 10.1002/eap.2043.

[12] K. A. N. and B. S. B Santoso, H. Wijayanto, "Synthetic Over Sampling Methods for Handling Class Imbalanced Problems: A Review," *Water (Switzerland)*, vol. 26, no. 2, pp. 1–72, 2017, doi: doi:10.1088/1755-1315/58/1/012031.

[13] K. He, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *Delving Deep into Rectifiers Surpassing Human-Level Perform. ImageNet Classif.*, 2015, doi: 10.1109/ICCV.2015.123.

[14] S. Sharma and S. Sharma, "Activation Functions in Neural Networks," vol. 4, no. 12, pp. 310–316, 2020.

[15] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *Adam A Method Stoch. Optim.*, pp. 1–15, 2015.

[16] A. Labach, H. Salehinejad, and S. Valaee, "Survey of Dropout Methods for Deep Neural Networks," 2019, [Online]. Available: <http://arxiv.org/abs/1904.13310>.

[17] T. D. K. Thara, P. S. Prema, and F. Xiong, "Auto-Detection of Epileptic Seizure Events Using Deep Neural Network with Different Feature Scaling Techniques," *Pattern Recognit. Lett.*, vol. 128, pp. 544–550, 2019, doi: 10.1016/j.patrec.2019.10.029.

[18] J. Hao, "Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language," vol. XX, no. X, pp. 1–14, 2019, doi: 10.3102/1076998619832248.

[19] A. Liu, J. Ghosh, and C. Martin, "Generative Oversampling for Mining Imbalanced Datasets," *Int. Conf. Data Min.*, pp. 25–28, 2007.

[20] A. Tharwat, "Classification Assessment Methods," *Appl. Comput. Informatics*, vol. 17, no. 1, pp. 168–192, 2018, doi: 10.1016/j.aci.2018.08.003.

[21] T. Djatna, M. Kusuma, D. Hardhienata, A. Fitri, and N. Masruriyah, "An intuitionistic Fuzzy Diagnosis Analytics for Stroke Disease," *J. Big Data*, 2018, doi: 10.1186/s40537-018-0142-7.

[22] M. Gordon and M. Kochen, "Recall-precision trade-off: A derivation," *J. Am. Soc. Inf. Sci.*, vol. 40, no. 3, pp. 145–151, 1989, doi: 10.1002/(SICI)1097-4571(198905)40:3<145::AID-ASII>3.0.CO;2-I.

[23] F. Ri *et al.*, "Effective Prediction of Three Common Diseases by Combining SMOTE with Tomek links Technique for Imbalanced Medical Data," vol. 2016, pp. 225–228, 2016, doi: 10.1109/ICOACS.2016.7563084.