Klasifikasi Human Stress Menggunakan Adagrad Optimization untuk Arsitektur Deep Neural Network

Mochammad Abdul Azis*[1], Ahmad Fauzi[2], Ginabila[3], Imam Nawawi[4]
[1] Program Studi Teknologi Informasi Universitas Bina Sarana Informatika; Jl. Kramat Raya No.98, Senen,Jakarta Pusat 10450, telp. (021)23231170
[2,3,4] Program Studi Sistem Informasi Universitas Bina Sarana Informatika; Jl. Kramat Raya No.98, Senen,Jakarta Pusat 10450, telp. (021)23231170
Email: [1] mochamad.mmz@bsi.ac.id, [2] ahmad.aau@bsi.ac.id, [3] gina.gnb@bsi.ac.id, [4] imam.imw@bsi.ac.id,

Abstrak

Menurut Organisasi Kesehatan Dunia, stres adalah jenis penyakit jiwa yang mempengaruhi kesehatan bagi manusia dan tidak ada seorang pun di dunia ini yang tidak menderita stres atau depresi, Stres merupakan sebutan yang kerap digunakan secara persamaan kata(sinonim) dengan pengalaman hidup ataupun kejadian hidup yang negative. Analisis data yang memiliki kelas tidak seimbang mengakibatkan ketidakakuratan dalam memprediksi human stress. Penelitian ini menunjukan bahwa menggunakan model Arsitektur Deep Neural Network (DNN) dengan melakukan optimasi pada beberapa parameter yaitu oprimizer, Learning rate dan epoch. Hasil Arsitekrut DNN paling baik didapatkan dengan 4 Hidden Layer, Optimasi Adagard, Learning rate 0.01 dan jumlah epoch 100. Skor akurasi, presisi, recall dan f-measure masing-masing mendapatkan 98.25%, 83.00%, 98.25%, 91.00%.

Kata kunci*— DNN, Klasifikasi, Stress, Optimasi, Kesehatan*

Abstract

According to the World Health Organization, stress is a type of mental illness that affects human health and there is no one in this world who does not suffer from stress or depression. Stress is a term that is often used synonymously with negative life experiences or life events. . Analysis of data that has an unbalanced class results in inaccuracies in predicting human stress. This study shows that using the Deep Neural Network (DNN) Architecture model by optimizing several parameters, namely the optimizer, Learning rate and epoch. The best DNN Architect results are obtained with 4 Hidden Layers, Adagard Optimization, Learning rate 0.01 and the number of epochs 100. Accuracy, precision, recall and f-measure scores get 98.25%, 83.00%, 98.25%, 91.00%, respectively.

Keywords*—DNN, Classification, Stress, Optimization, Health*

1. PENDAHULUAN

Di dunia saat ini, salah satu faktor utama penyebab ketidaksehatan adalah Stres. Stres merupakan sebutan yang kerap digunakan secara persamaan kata(sinonim) dengan pengalaman hidup ataupun kejadian hidup yang negatif. Menurut Organisasi Kesehatan Dunia, stres adalah jenis penyakit jiwa yang mempengaruhi kesehatan bagi manusia dan tidak ada seorang pun di dunia ini yang tidak menderita stres atau depresi.

Stres yang berkelanjutan dapat menyebabkan masalah medis yang nyata termasuk kegelisahan, kurang tidur, nyeri otot, hipertensi dan sistem kekebalan tubuh yang lemah (Malathi & Thiyagarajan, 2020). Karena stres mungkin ada masalah kesehatan lain seperti obesitas, serangan jantung, diabetes, asma dan lain sebagainya (Ahuja & Banga, 2019).

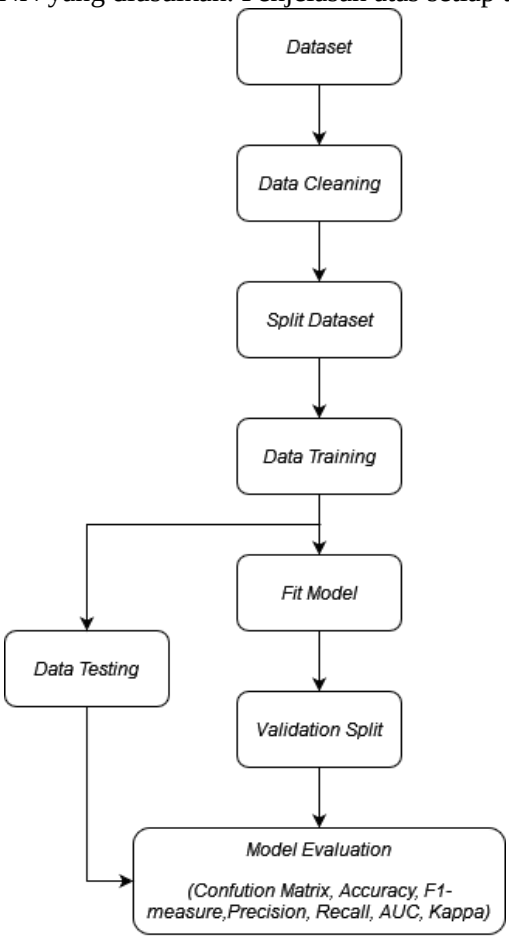
Ini adalah penyakit saraf mental yang umum, Hal ini dapat dibedakan sebagai perasaan sedih, kehilangan, jengkel yang menghambat perilaku sehari-hari seseorang. Orang mengalami berbagai bentuk depresi Itu bisa mengganggu pekerjaan sehari-hari (Biilah et al., 2022), Ada tiga jenis stres tergantung pada periode waktu di mana stres terjadi, dan jenis ini adalah: stres kronis, stres akut episodik, dan stres akut. Sedangkan stres kronis tinggal untuk waktu yang lama, stres akut menghilang cepat (Seaward, 2018).

Pada stres kronis, sel-sel tubuh menjadi tidak peka terhadap ortisol, yang menyebabkan

peradangan dan menyebabkan kerusakan sel otak dan pembuluh darah (Elzeiny & Qaraqe, 2019), Pada penelitian lainnya yang dilakukan dengan judul "Human Stress Detection With Wearable Sensors Using Convolutional Neural Networks" Algoritma klasifikasi yang digunakan untuk memprediksi Human Stress yaitu deep learning architecture dengan convolutional neural networks dalam penelitian ini CNN memberikan akurasi sebesar 96,6% (Gil-Martin et al., 2022).

2. METODE PENELITIAN

Penelitian dilakukan sesuai dengan tahapan-tahapan sebagaimana disajikan pada Gambar 1. Penelitian dimulai dari melakukan identifikasi objek permasalahan sampai dengan melakukan evaluasi model arsitektur DNN yang diusulkan. Penjelasan atas setiap tahapan sebagai berikut.



Gambar 1. Tahapan Penelitian

Gambar diatas menjelaskan mengenai tahapan-tahapan yang dilakukan oleh peneliti. Tahapan pertama merupakan tahap pengumpulan dataset untuk keperluan penelitian Tahapan persiapan data disini juga dilakukan untuk memastikan bahwa *dataset* yang digunakan untuk *Training*, *Testing* dan *Validation* merupakan data yang berkualitas. Jika *dataset* yang digunakan masih memiliki *noise* maka model yang dihasilkan juga tidak akan berkualitas dan memiliki *bias*. Lalu kemudian dataset tersebut melalui proses data cleaning yang digunakan untuk memastikan kebenaran, konsistensi dan kegunaannya dalam penelitian. *Data cleaning* disini bertujuan untuk menghapus *noise*, data yang tidak konsisten dan adanya kesalahan pada *dataset*. Salah satu bentuk *noise* adalah terjadinya nilai yang hilang (*missing values*). Salah satu teknik yang dapat digunakan untuk menangani missing values adalah dengan melakukan *imputation*. Setelah peneliti melakukan data cleaning kemudian split dataset dilakukan untuk mengkasilkan data training. Kemudian data testing akan dilakukan. Selanjutnya ada tahap fit model, validation split dan terakhir menetapkan model evaluation.

2.1 Dataset

Dataset yang digunakan dalam penilitian ini memiliki 1 atribut *binary* yaitu S*tress Level*,

juga 3 atribut dalam bentuk numerik yaitu *Humidity*, *Temperature*, *Step count* dan *Stress Level* . Disini atibut binary memiliki dua kategori yaitu angka 0 atau angka 1, di mana angka 0 menjelaskan bahwa atribut itu tidak ada, dan angka 1 menjelaskan atribut itu ada. Atribut Biner disebut sebagai Boolean jika dalam prosesnya dinyatakan dengan benar (true) dan salah (false).

2.2 Data *Cleaning*

Pembersihan Data berarti proses mengidentifikasi bagian data yang tidak benar, tidak lengkap, tidak akurat, tidak relevan, atau hilang dan kemudian memodifikasi, mengganti, atau menghapusnya sesuai kebutuhan. Pembersihan data dianggap sebagai elemen dasar dari ilmu data dasar. Pada langkah selanjutnya, setelah menganalisis dan menggali dataset maka kita harus membersihkan data. Pada proses pembersihan ini semua nilai duplikat dan nilai null yang ada pada dataset akan dihilangkan dan akan didapatkan dataset baru (Sudha dan Akila, 2020)

Kegiatan data cleaning akan sangat tergantung dari hasil identifikasi permasalahan dengan Exploratory Data Analysis (EDA). Pada dataset yang sering terjadi adalah adanya missing values yang harus ditangani. Teknik imputasi terbagi menjadi dua yaitu Statistical Techniques dan Machine Learning Techniques (Tsai dan Lin, 2022). Ada beberapa metode pada Teknik Statistical Techniques yang dapat digunakan untuk menangani missing values, yaitu dengan melakukan imputasi menggunakan nilai rata-rata (mean), imputasi dengan nilai tengah (median) atau dengan nilai yang memiliki frekuensi paling sering (mode).

2.3 *Label Encoding*

Dalam banyak kegiatan *machine learning* atau *data science*, kumpulan data mungkin berisi teks atau nilai kategorikal (pada dasarnya nilai non-numerik). Misalnya, fitur warna yang memiliki nilai seperti kuning, oranye, hijau, putih, dll. Paket makan memiliki nilai seperti sarapan, makan siang, camilan, makan malam, kopi, dan lain-lain. Beberapa algoritma seperti dapat menangani nilai kategoris dengan sangat baik, tetapi sebagian besar dari algoritma mengharapkan nilai numerik untuk mencapai hasil yang lebih baik. Ada banyak cara untuk mengubah nilai kategorikal menjadi nilai numerik.

2.4 *Feature Scalling*

Standardisasi atau *feature scaling* merupakan salah satu tahapan selama data *pre-processing.* Hal ini dilakukan untuk menormalkan data dalam rentang tertentu. *Feature scaling* bertujuan untuk membantu dalam mempercepat perhitungan dalam algoritma. *Feature scaling* data adalah persyaratan umum untuk eksperimen yang dilakukan menggunakan Keras, *Scikit-learn* dan *Deep Learning*. Ada banyak scaler berbeda yang tersedia untuk feature scaling dataset. Yang paling umum digunakan adalah *Normalizer, RobustScaler, MinMaxScaler,* dan *StandardScaler*. (Thara et al., 2019).

2.5 Pembuatan Model Arsitektur DNN

Model Arsitektur DNN dibuat berdasarkan jumlah hidden layer yang teridiri 4 *hidden layer*. Penentuan jumlah *hidden layer* menggunakan Keras Tuner dan teknik *random search*, karena jumlah parameter yang dipertimbangkan sangat tinggi dan besarnya pengaruh tidak seimbang. Untuk mengatasi bias yang dihasilkan karena proses *random search* diperlukan *cross validation.* Pada beberapa implementasi memerlukan jumlah layer yang lebih banyak untuk meningkatkan kinerja model (Khalil et al., 2018). Eksperimen dilakukan dengan beberapa variasi *setting hyperparameter*.

Tabel 1. Arsitektur DNN

Input Neuron	***Hidden Layer***	***Hidden Neuron***
3	4	32,64,128,256

Eksperimen akan dilakukan dengan mengimplementasikan teknik *features extraction.* Arsitektur model DNN dan jumlah neuron pada setiap *hidden layer* sebagaimana disajikan pada Tabel 1. Arsitektur model DNN memiliki input neuron sebanyak 3 sesuai dengan jumlah *feature predictor* pada dataset.

Penentuan jumlah neuron pada *hidden layer* dilakukan secara random dengan pola dari besar-kecil-besar-kecil dengan menggunakan kelipatan angka 32 (Invernizzi et al., 2021). Semua

hidden layer pertama dimulai dengan jumlah neuron 480 yang merupakan perkalian dari jumlah input neuron dengan kelipatan yang digunakan untuk penentuan jumlah neuron pada *hidden layer* dan semua neuron antar *layer fully connected* (Khalil et al., 2018).

Fungsi aktivasi pada *input layer* dan *hidden layer* menggunakan ReLu., sedangkan untuk *output layer* menggunakan Sigmoid. Eksperimen masing- masing akan dilakukan dengan jumlah epoch 100 dan batch size 50. Variasi Model Deep Neural Network (DNN) dengan parameter dan nilainya sebagaimana pada Tabel 2.

Tabel 2. Parameter Model DNN

Parameter	**Nilai Parameter**
Aktivasi Input Layer	*ReLU (Rectified Linear Unit)*
Aktivasi Hidden Layer	*ReLU (Rectified Linear Unit)*
Aktivasi Output	*Sigmoid*
Epoch	100
Batch Size	50

Kombinasi parameter optimasi Model DNN sebagaimana disajikan pada Tabel 3. Berdasarkan komposisi pada tabel tersebut, eksperimen akan dilakukan dengan optimasi Adagrad dengan nilai learning rate sebesar 0,01. Pemodelan dilakukan dengan menggunakan keras dan *back-end tensorflow*.

Tabel 3. Parameter Optimasi Model DNN

Optimasi	***Learning Rate***
AdaGrad	0.01

2.6 Eksperimen dan Validasi

Eksperimen dilakukan menggunakan aplikasi Google Colabs dengan Bahasa pemrogaman Python 3.8.0. Spesifikasi komputer yang digunakan untuk running program sebagaimana disajikan pada Tabel 4.

Tabel 4. Spesifikasi Komputer

Computer Spesification	
Sistem Operasi	*Windows* 10 *Pro 64-Bit Version 20H2 (Build 19042.1110)*
Processor	*Intel(R) Core(TM) i5-9400T CPU @ 1.80GHz 1.80 GHz*
RAM	8GB DDR4 (2666MHz)
Harddisk	1TB 5400RPM 3.5" SATA III + 256GB NVMe M.2 SSD
Aplikasi *Data Mining*	*Google Colabs*

Teknik validasi model DNN yang diimplementasikan pada penelitian ini menggunakan *validation split*. Prinsip kerja *validation split* membagi data training menjadi dua bagian. Bagian pertama sebagai data training dan bagian akhir sebagai data validasi (*hold-out to validate on while training*).

3. HASIL DAN PEMBAHASAN

3.1 *Exploratory Data Analysis* (EDA)

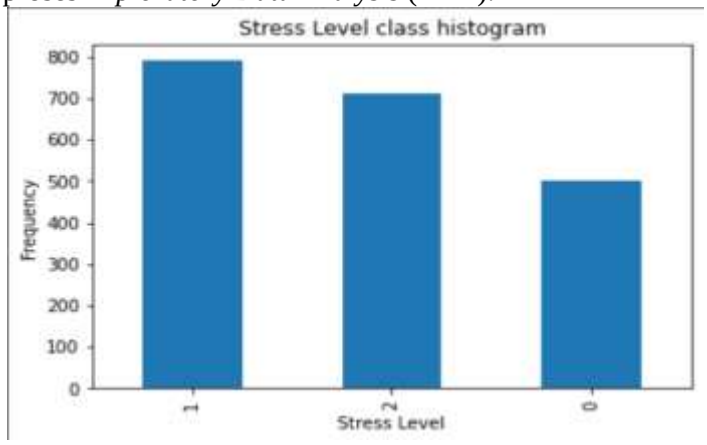
Tahap pertama yang dilakukan dalam eksperimen adalah menganalisis statistik dataset dengan melakukan proses *Exploratory Data Analysis* (EDA).

Tabel 5. *Feature Important Coefisien*

Feature	*Important Coefisien*
Humidity	0.341384
Temperature	0.463540
Step count	0.195077

3.2 *Imbalance Class*

Tahap pertama yang dilakukan dalam eksperimen adalah menganalisis statistik dataset dengan melakukan proses *Exploratory Data Analysis* (EDA).



Gambar 2. *Class Target Distribution*

Berdasarkan gambar 2 diatas dapat diketahui bahwa frekuensi tertinggi dari stress level mencapai angka 800 dan yang terendah frekuensi yang didapat hanya mencapai angka 500.

3.3 Hasil Eksperimen

Eksperimen pada setiap model dilakukan secara bertahap berdasarkan teknik *Split Dataset* yang diimplementasikan. Penggujian terhadap *hyperparameter* dilakukan dengan menggunakan 4 *hidden layer*, 2 variasi fungsi aktivasi yaitu pada setiap *hidden layer* digunakan fungsi aktivasi ReLU dan pada output layer digunakan fungsi aktivasi Sigmoid. Kemudian menggunakan optimizer Adagrad dan *learning rate* 0,01. Model ini di-training sebanyak 100 epoch dan *batch size* 50.

Hasil analisa berdasarkan model yang paling baik untuk setiap *metric performance*. Pada Tabel 6. merupakan hasil *accuracy score* berdasarkan optimasi. Data pada tabel tersebut dapat dianalisis bahwa *accuracy score* terbaik didapatkan oleh optimasi Adagrad, jumlah epoch 100 dan optimasi *learning rate* 0.01. *Accuracy Score* yang didapatkan optimasi Adagrad sebesar 98.25%. Hal tersebut menunjukkan bahwa dari hasil eksperimen dengan optimasi Adagrad dan *learning rate* yang sangat tinggi dibandingkan optimisi lainnya.

Tabel 6. Rekapitulasi Perbandingan *Accuracy Score*

Optimasi	*Epoch*	*LR*	*Accuracy*
AdaGrad	100	0.01	98.25%

Pada Tabel 7. merupakan rekapitulasi hasil *recall score* berdasarkan kombinasi optimasi dapat dianalisis bahwa *score* terbaik didapatkan oleh optimasi Adagrad dengan jumlah *epoch* 100. Untuk setting nilai *learning rate* optimasi Adagrad sebesar 0.01. *Recall score* yang didapatkan optimasi Adagrad sebesar 98,25%.

Tabel 7. Rekapitulasi Perbandingan *Recall Score*

Optimasi	*Epoch*	*LR*	*Recall*
AdaGrad	100	0.01	98.25%

Pada Tabel 8 merupakan rekapitulasi perbandingan hasil *precision score* berdasarkan kombinasi optimasi dapat dianalisis bahwa *score* terbaik optimasi Adagrad dengan jumlah *epoch* 100. Untuk setting nilai *learning rate* optimasi Adagrad sebesar 0.01. *Precision score* yang didapatkan model A dan C sebesar 93.00%.

Tabel 8. Rekapitulasi Perbandingan *Precision Score*

Optimasi	*Epoch*	*LR*	*Precision*
AdaGrad	100	0.01	83,00%

Pada Tabel 9. merupakan rekapitulasi hasil *F1-score* berdasarkan optimasi Adagrad. Data pada tabel tersebut dapat dilihat bahwa *score* terbaik didapatkan dengan optimasi Adagrad dan jumlah *epoch* 100. Untuk setting nilai *learning rate* optimasi Adagrad sebesar 0.01. *F1-score* yang didapatkan optimasi Adagrad sebesar 91.00%.

Tabel 9. Rekapitulasi Perbandingan *F1-Score*

Optimasi	*Epoch*	*LR*	*F1-Score*
AdaGrad	100	0.01	91.00%

Pada Tabel 10. merupakan hasil nilai AUC berdasarkan optimasi Adagrad dapat nilai AUC terbaik sebesar 0.982. Untuk *learning rate* optimasi Adagrad sebesar 0.01. Nilai AUC ini sangat dipengaruhi *True Positive Rate* dan *False Positive Rate*.

Tabel 10. Rekapitulasi Perbandingan Nilai AUC

Optimasi	*Epoch*	*LR*	*AUC*
AdaGrad	100	0.01	0.982

Berdasarkan observasi pada Tabel 11, score untuk accuracy dan f-measure didapatkan oleh optimasi Adagrad, learning rate 0.01, jumlah epoch 100. Sehingga model yang dipilih sebagai model prediktif Human Stress.

Tabel 11. Rekapitulasi *Score* Tertinggi *Metric Performance*

Metric Performance	*Score*
Accuracy	98.25%
Precision	83,00%
Recall	98,25%
F1-Score	91,00%
AUC	0,982

3.4 Arsitektur DNN dan Nilai Parameter

Berdasarkan analisis hasil eksperimen model yang dipilih untuk diusulkan sebagai model prediktif adalah optimasi Adagrad, learning rate 0.01. Parameter Arsitektur model DNN yang dipilih sebagaimana disajikan pada Tabel 12. dengan jumlah *input neuron* sebanyak 3 unit sesuai dengan *feature dataset*. Jumlah *hidden layer* sebanyak 4 dengan jumlah *neuron* masing-masing layer 32, 64, 128, 256 unit. Fungsi Aktivasi *input layer* dan *hidden layer* menggunakan *ReLU*, sedangkan untuk *output layer* menggunakan *Sigmoid*. Optimasi yang digunakan Adagrad dengan learning rate 0.01.

Tabel 12. Parameter Arsitektur DNN yang Diusulkan

Parameter	**Nilai Parameter**
Input Neuron	3
Hidden Layer	4
Hidden Neuron	32,64 ,128,256
Epoch	100
Batch Size	50
Fungsi Aktivasi *Input* dan *Hidden Layer*	*ReLU*
Fungsi Aktivasi *Output Layer*	*Sigmoid*
Optimasi	Adagrad (lr = 0.01)

4. KESIMPULAN

Berdasarkan analisis dan pembahasan hasil penelitian, dapat disimpulkan bahwa:

a. Arsitektur model DNN dapat digunakan untuk memprediksi Human Stress.
b. Penggunaan berbagai variasi optimasi pada model DNN mendapatkan *score metrics performance* yang berbeda-beda.

c. Tingkat akurasi paling baik sebesar 98,25% didapatkan arsitektur:
   1) Model DNN
      a) Jumlah *Input Layer* sebanyak 3.
      b) Jumlah *Hidden Layer* sebanyak 4 dengan 32, 64, 128, 256 *neuron*.
      c) Jumlah *Output Layer* sebanyak 1.
   2) Parameter Model DNN
      a) Aktivasi *Input Layer* dan *Hidden Layer* menggunakan *ReLU*.
      b) Aktivasi *Output Layer* menggunakan *Sigmoid*.
      c) Jumlah *epoch*: 100.
      d) *Batch size*: 50
   3) Parameter optimasi model DNN menggunakan Adagrad dengan *learning rate* 0.01.

DAFTAR PUSTAKA

[1] Ahuja, R., & Banga, A. (2019). Mental stress detection in university students using machine learning algorithms. *Procedia Computer Science*, *152*, 349–353. https://doi.org/10.1016/j.procs.2019.05.007

[2] Biilah, M., Raihan, M., Akter, T., Alvi, N., Bristy, N. J., Rehana, H., & others. (2022). Human Depression Prediction Using Association Rule Mining Technique. *International Conference on Innovative Computing and Communications*, 223–237.

[3] Elzeiny, S., & Qaraqe, M. (2019). Machine Learning Approaches to Automatic Stress Detection: A Review. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, *2018-Novem*, 1–6. https://doi.org/10.1109/AICCSA.2018.8612825

[4] Gil-Martin, M., San-Segundo, R., Mateos, A., & Ferreiros-Lopez, J. (2022). Human Stress Detection With Wearable Sensors Using Convolutional Neural Networks. *IEEE Aerospace and Electronic Systems Magazine*, *37*(1), 60–70.

[5] Invernizzi, L., Long, J., Chollet, F., O'Malley, T., & Jin, H. (2021). *Getting Started with KerasTuner Start the Search*. https://keras.io/guides/keras_tuner/getting_started/

[6] Khalil, K., Eldash, O., Kumar, A., & Bayoumi, M. (2018). An Efficient approach for neural network architecture. *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 745–748.

[7] Malathi, V., & Thiyagarajan, A. (2020). Prediction of Ground Water Level based on machine Learning. *Rit.Edu*. https://www.rit.edu/kgcoe/brainlab/sites/rit.edu.kgcoe.brainlab/files/IGSC_2021_paper_4.pdf

[8] Seaward, B. (2018). *Managing Stress: Principles and Strategies for Health and Well-Being, Edisi 9*. Boston: Jones \& Bartlett. https://books. google. co. id/[diakses pada 28~….

[9] Thara, D. K., PremaSudha, B. G., & Xiong, F. (2019). Auto-Detection of Epileptic Seizure Events Using Deep Neural Network with Different Feature Scaling Techniques. *Pattern Recognition Letters*, *128*, 544–550.

[10] Tsai, C.-F., Li, M.-L., & Lin, W.-C. (2018). A Class Center Based Approach for Missing Value Imputation. *Knowledge-Based Systems*, *151*, 124–135.

[11] C. Sudha and D. Akila, "Credit Card Fraud Detection Using AES Technic," pp. 91–98, 2020, doi: 10.1007/978-981-15-3284-9.

[12] C.-F. Tsai, M.-L. Li, and W.-C. Lin, "A Class Center Based Approach for Missing Value Imputation," *Knowledge-Based Syst.*, vol. 151, pp. 124–135, 2018, doi: 10.1016/j.knosys.2018.03.026.