

# Klasifikasi Tingkat Stres Mahasiswa Menggunakan RMSProp untuk Arsitektur *Artificial Neural Network*

Satria Baladewa Harahap<sup>1</sup>, Yuni Yamasari<sup>2</sup>

<sup>1,2</sup> Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya

<sup>1</sup>[satriabaladewa.20080@mhs.unesa.ac.id](mailto:satriabaladewa.20080@mhs.unesa.ac.id)

<sup>2</sup>[yuniyamasari@unesa.ac.id](mailto:yuniyamasari@unesa.ac.id)

**Abstrak**— Beratnya berbagai tuntutan yang dihadapi mahasiswa membuat mereka kerap mengalami gejala stres. Gejala stres yang terus menerus dapat memperburuk keadaan psikis dan fisik mahasiswa yang pada akhirnya menyebabkan turunnya kinerja akademik mereka. Namun, gejala stres tidak dapat ditangani secara sembarangan. Perlu adanya penanganan gejala stres yang tepat yang disesuaikan dengan tingkat stres yang dialami. Untuk itu, pengklasifikasian tingkat stres mahasiswa sangat diperlukan untuk memberikan penanganan lebih lanjut yang tepat. Oleh karena itu, penelitian ini memfokuskan pada bagaimana cara mengklasifikasikan tingkat stres mahasiswa secara otomatis menggunakan teknologi Artificial Neural Network (ANN). Dalam penelitian ini, kinerja ANN ditingkatkan dengan melakukan pengujian terhadap RMSProp optimizer dengan berbagai variasi hyperparameter dan percobaan dengan dan tanpa oversampling SMOTE-N. Setelah pengujian dilakukan sebanyak 52 kali, hasil ujicoba memperlihatkan bahwa RMSProp dengan SMOTE-N lebih unggul dibanding RMSProp tanpa SMOTE-N. Akurasi RMSProp dengan SMOTE-N berhasil mencapai 96%, sedangkan akurasi RMSProp tanpa SMOTE-N sebesar 88% sehingga SMOTE-N pada model dapat meningkatkan kinerja sebesar 8%.

**Kata Kunci**— klasifikasi, stres, mahasiswa, ANN, SMOTE-N

## I. PENDAHULUAN

Kesehatan merupakan keadaan dimana fisik dan mental seseorang dalam keadaan sejahtera dan tanpa adanya kelemahan sehingga seseorang tersebut dapat melakukan produktivitas baik secara sosial maupun ekonomis. Jika kesejahteraan ini tidak terpenuhi, maka timbul masalah kesehatan, baik itu kesehatan fisik maupun psikologis. Gangguan psikologis salah satunya adalah stres [1]. Stres adalah keadaan yang disebabkan oleh tuntutan fisik maupun lingkungan, gangguan sehari-hari, dan perubahan dalam hidup yang memerlukan usaha dan penyesuaian secara terus menerus. Stres dapat dibagi menjadi tiga tingkatan, yaitu stres ringan, sedang, dan tinggi [2]. Stres dapat dialami oleh siapapun, termasuk mahasiswa [3].

Di Indonesia, persentase mahasiswa mengalami stres mencapai angka yang cukup tinggi, yaitu 36,7%-71,6% [4]. Menurut hasil wawancara yang dilakukan oleh Merry dan Henny Christine Mamahit pada beberapa mahasiswa di salah satu perguruan tinggi di Indonesia, beberapa diantaranya mengalami bermacam-macam gejala stres, seperti susah tidur hingga menggunakan rokok sebagai media untuk merileksasi pikiran. Beberapa dari mereka menyatakan bahwa mata kuliah yang ditawarkan begitu banyak materi yang perlu dipelajari

dengan tingkat kesulitan yang cukup tinggi. Hal tersebut membuat mereka menjadi tidak yakin mampu menyelesaikan mata kuliah yang diambil dan cenderung menghindari mata kuliah yang sulit. Terlebih saat ujian, tidak ada sistem remedial sehingga menyebabkan nilai mereka cenderung rendah dan hal tersebut menyebabkan perasaan cemas pada diri mahasiswa [4].

Mayoritas mahasiswa mengalami stres yang disebabkan oleh banyaknya tanggung jawab, seperti pengerjaan tugas kuliah atau bahkan skripsi bagi mahasiswa tingkat akhir. Selain itu, stres pada mahasiswa juga dipicu oleh berbagai tuntutan, baik itu tuntutan ekonomi maupun tuntutan sosial. Mahasiswa tingkat akhir sering dituntut orang tua mereka untuk segera wisuda [2]. Adapun mahasiswa tahun pertama yang mengalami *homesick* dikarenakan mereka berada di lingkungan yang baru sehingga menuntut mereka untuk dapat beradaptasi dengan lingkungan sekitar [5]. Hal-hal tersebut yang membuat mahasiswa merasa terbebani dan akhirnya berujung stres. Apabila mahasiswa terus menerus mengalami stres, maka akan menyebabkan penurunan hasil akademis. Stres yang terlalu berat dapat mengganggu konsentrasi dan kinerja mahasiswa [3].

Menurut Salu, stres dalam jangka waktu yang pendek akan menyebabkan seseorang mudah untuk marah dan kesulitan berkonsentrasi. Namun, apabila tidak segera diatasi, maka dapat menyebabkan risiko jangka panjang, yaitu beberapa gangguan seperti gangguan otak, kesehatan, sistem saraf dan imun [6]. Dengan melihat begitu besarnya dampak yang ditimbulkan oleh stres yang dialami mahasiswa, maka perlu adanya tindakan yang dapat menjadi solusi untuk mengurangi tingkat stres yang terjadi pada mahasiswa. Namun, tiap tingkatan stres memiliki penanganan yang berbeda dikarenakan tiap tingkatan stres menghasilkan gejala psikis maupun fisik yang berbeda pula [4]. Oleh karena itu, perlu adanya sistem yang dapat mengklasifikasikan tingkat stres mahasiswa agar mahasiswa mendapatkan edukasi yang tepat mengenai penanganan dari gejala stres berdasarkan tingkat stres yang dialaminya supaya mahasiswa dapat segera terhindar dari dampak stres berkepanjangan. Di dalam dunia psikologi saat ini, telah ada beberapa penelitian yang mengklasifikasikan tingkat stres. Salah satu contohnya adalah penelitian Seto dkk [2] yang mengklasifikasikan tingkat stres pada mahasiswa dan mencari keterkaitannya dengan tingkat motivasi belajar pada mahasiswa dengan menggunakan uji *Pearson Correlation*. Namun seiring berkembangnya zaman, sistem klasifikasi stres dapat dilakukan secara otomatis menggunakan teknologi. Sistem inilah yang akan diusulkan pada penelitian ini yang akan menggunakan teknologi *data mining*.

*Data mining* merupakan pilihan yang tepat untuk mengklasifikasikan tingkat stres. Namun, saat ini belum ada sistem yang dapat digunakan oleh masyarakat luas untuk membantu mengklasifikasikan tingkat stres [7]. Oleh karena itu, penelitian ini akan mengusulkan pembuatan model untuk klasifikasi tingkat stres mahasiswa. Nantinya, model ini dapat digunakan oleh para pengembang untuk membuat sistem yang dapat diakses oleh mahasiswa dan masyarakat luas untuk memprediksi tingkat stres yang dialaminya dan mendapat edukasi yang tepat tentang cara penanganan stres sesuai tingkat stresnya.

Penelitian serupa telah dilakukan oleh Anjani & Yamasari [8] pada penelitiannya yang berjudul “Klasifikasi Tingkat Stres Mahasiswa Menggunakan Metode Berbasis *Tree*.” Penelitian ini mengklasifikasikan *dataset* tingkat stres pada mahasiswa Universitas Negeri Surabaya dengan berbagai metode *Tree*, yaitu *Random forest*, *Random tree*, CHAID (*Chi-squared Automatic Interaction Detector*), *Decision tree*, ID3 (*Iterative Dichotomizer 3*), dan *Decision stump*. Melalui penelitian ini, didapatkan bahwa metode ID3 meraih akurasi tertinggi pada perbandingan *dataset* 80:20, yaitu sebesar 95%. Namun, pada penelitian lain yang membandingkan antara metode *Iterative Dichotomizer 3* (ID3), *Artificial Neural Network* (ANN), *Naïve Bayes*, dan *Support Vector Machine* (SVM), didapatkan kesimpulan bahwa ANN lebih unggul dibanding metode lainnya termasuk ID3, dengan perolehan akurasi 93% ANN, 85% ID3, 69% *Naïve Bayes*, dan 87% SVM [9].

Oleh karena itu, penelitian ini tertarik untuk menggunakan *Artificial Neural Network* (ANN) untuk mengklasifikasikan *dataset* tingkat stres mahasiswa yang dimiliki oleh Anjani & Yamasari [8] pada penelitiannya. Turunan dari ANN yang akan digunakan di dalam penelitian ini adalah *Multi Layer Perceptron* (MLP). MLP merupakan jaringan yang terhubung sepenuhnya (*fully connected*) yang terdiri dari sebuah *input layer*, satu atau lebih *hidden layer*, dan *output layer* yang terinspirasi dari cara kerja otak manusia [10]. MLP dipilih karena algoritma ini relatif sederhana dan dapat memberikan hasil akurasi yang tinggi dengan waktu pelatihan yang lebih cepat ketimbang algoritma klasifikasi lainnya yang memiliki arsitektur yang lebih kompleks [11]. Pernyataan ini dibuktikan oleh penelitian Samarpita dkk [12] dalam penelitiannya yang melakukan klasifikasi tingkat stres menggunakan MLP dan mendapatkan akurasi yang cukup tinggi, yaitu sebesar 95%.

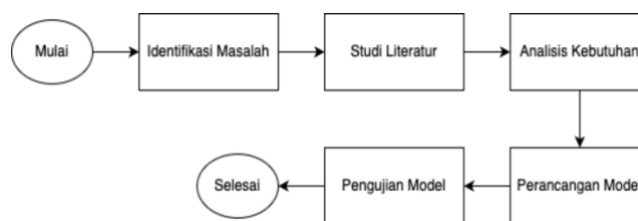
Untuk mengoptimalkan performa ANN, penelitian ini menggunakan suatu *optimizer* yang bernama RMSProp. Penelitian sebelumnya yang dilakukan oleh Desai [13] menggunakan *neural network* dengan RMSProp. Dari penelitian tersebut, model dengan RMSProp terbukti mampu untuk mendapatkan performa yang sangat baik, yaitu tingkat akurasi yang mencapai 95,05%. *Optimizer* ini nantinya akan diukur performanya menggunakan beberapa metrik evaluasi, yaitu *accuracy*, *precision*, *recall*, *F1-score*, dan *Area Under The Curve*(AUC).

Dengan optimalisasi yang dilakukan, diharapkan terbentuk model yang andal dan dapat membantu pengembangan sistem

di masa depan yang dapat diakses oleh masyarakat luas khususnya mahasiswa untuk mengklasifikasikan tingkat stresnya secara akurat agar mereka mendapat edukasi penanganan stres yang tepat sesuai dengan tingkatan stres yang dialaminya agar mahasiswa terhindar dari berbagai macam dampak dari stres berkepanjangan.

## II. METODE PENELITIAN

Untuk memudahkan dalam mengetahui proses pelaksanaan suatu penelitian, perlu untuk dibuat diagram alur penelitian. Alur pada penelitian ini dimulai dari tahap identifikasi masalah yang akan dipecahkan hingga sampai pada pengujian model *Artificial Neural Network* (ANN). Diagram alur pada penelitian ini dapat dilihat pada Gbr. 1.



Gbr. 1 Alur Penelitian

### A. Identifikasi Masalah

Permasalahan pada penelitian ini adalah banyaknya mahasiswa yang mengalami stres. Mahasiswa perlu mendapatkan edukasi penanganan gejala stres agar tidak mengalami gejala stres berkepanjangan. Namun, tiap tingkatan stres memiliki penanganan yang berbeda pula. Oleh karena itu, perlu adanya klasifikasi tingkat stres agar mahasiswa dapat diberikan edukasi penanganan gejala stres sesuai dengan tingkat stres yang dialaminya. Namun, saat ini belum ada pengklasifikasi stres yang dapat digunakan oleh masyarakat luas sehingga penelitian ini ingin memberikan solusi dengan mengklasifikasikan tingkat stres mahasiswa dengan menggunakan ANN. Metode ini akan dioptimalisasi menggunakan *optimizer* RMSProp dengan dan tanpa SMOTE-N yang kemudian didapatkan hasil terbaiknya akan digunakan di dalam sistem untuk mengetahui tingkat stres mahasiswa secara akurat.

### B. Studi Literatur

Langkah selanjutnya, yaitu melakukan studi literatur. Studi literatur dilakukan dengan mempelajari teori mengenai *machine learning*, ANN, MLP, SKCV, SMOTE-N, dan RMSProp *optimizer*. Teori-teori tersebut didapatkan dari beberapa jurnal yang menjadi referensi dari penulisan penelitian ini. Jurnal yang didapat juga bermanfaat sebagai rujukan dalam menyelesaikan masalah pada penelitian yang diusulkan.

### C. Analisis Kebutuhan

Dalam melakukan penelitian, perlu untuk melakukan analisis kebutuhan untuk menunjang proses penelitian dari awal hingga akhir. Dalam penerapan metode ANN pada

penelitian ini, beberapa kebutuhan yang diperlukan diantaranya adalah Google Colab dan Python. Google Colab adalah *platform* yang memungkinkan penyimpanan, penulisan, dan berbagi program melalui Google Drive. Keuntungan utamanya termasuk kemampuan berkolaborasi online, akses gratis GPU, dan fleksibilitas dalam menjalankan program *machine learning*. Fitur lainnya termasuk tidak perlu konfigurasi tambahan, akses mudah ke Google Drive dan GitHub, serta penggunaan *library* Python untuk analisis data dan *machine learning* [14]. Python adalah bahasa pemrograman serbaguna yang fokus pada keterbacaan kode. Dilengkapi dengan beragam fungsionalitas dari pustaka standar yang luas, didukung oleh komunitas besar, dan mendukung berbagai paradigma pemrograman. Python menonjol dengan manajemen memori otomatis dan aplikasi yang luas dalam pengembangan perangkat lunak dan data analisis [15].

#### D. Perancangan Model

Penelitian yang diusulkan ini akan dirancang dengan model yang menggunakan metode ANN dalam pengklasifikasiannya. Metode ini dioptimalisasi oleh RMSProp *optimizer* yang memiliki persamaan (1). RMSProp akan diuji coba dengan dua skenario, yaitu dengan menggunakan *oversampling* SMOTE-N dan tanpa SMOTE-N. Hasil optimalisasi akan dibandingkan hasilnya sesuai dengan metrik evaluasi yang akan digunakan. Selanjutnya akan dipilih satu percobaan terbaik yang akan digunakan dalam pengujian model. Diagram perancangan model dapat dilihat pada Gbr. 3. Dalam perancangan model, *dataset* yang digunakan adalah *dataset* tingkat stres mahasiswa Universitas Negeri Surabaya yang berhasil dikumpulkan oleh Anjani & Yamasari pada penelitiannya [8]. *Dataset* ini berjumlah 100 sampel dengan 10 fitur dan 1 label yang berisi klasifikasi tingkat stres mahasiswa, yaitu ringan, sedang, dan berat. Sepuluh fitur tersebut terdiri dari nomor peserta, pertanyaan 1, pertanyaan 2, hingga pertanyaan 8, dan jumlah skor pertanyaan. Pertanyaan 1 sampai 8 merupakan pertanyaan terkait kehidupan pribadi mahasiswa. Skor yang dapat diberikan di tiap pertanyaan adalah 4, yaitu 1 untuk sangat tidak setuju, 2 untuk tidak setuju, 3 untuk setuju, dan 4 untuk sangat setuju. Pertanyaan yang diajukan terdapat pada Gbr. 2.

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1 - \beta) \left( \frac{\partial C}{\partial w} \right)^2$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}} \frac{\partial C}{\partial w} \quad (1)$$

Dimana:

$E[g]$  = rata – rata bergerak dari kuadran gradien

$\beta$  = parameter rata – rata bergerak

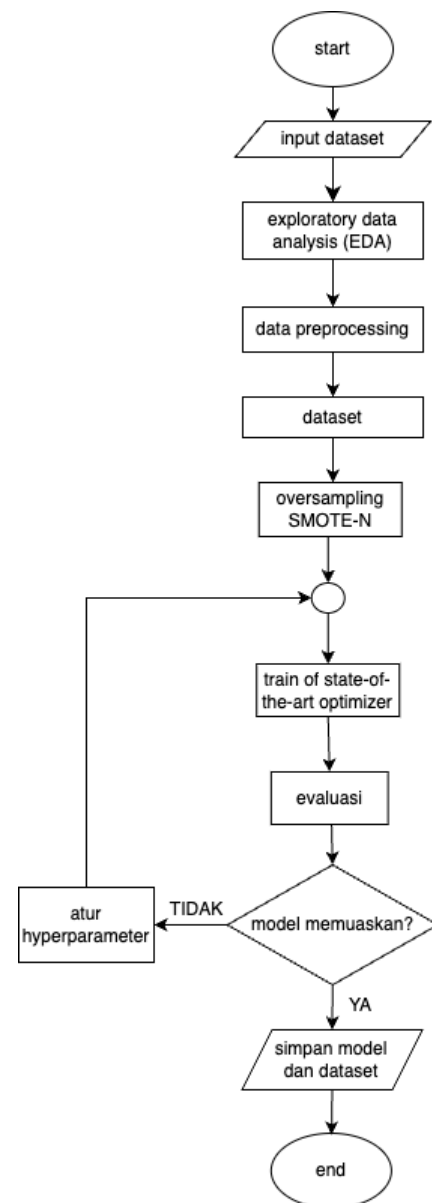
$\frac{\partial C}{\partial w}$  = gradien dari fungsi biaya terhadap bobot

$w$  = bobot

$\eta$  = laju pembelajaran

Pertanyaan 1 : Saya merasa tertekan oleh tuntutan orang tua yang menyuruh saya segera wisuda.
Pertanyaan 2 : Teman saya selalu mengajak saya bermain sehingga saya merasa tidak fokus dalam mengerjakan tugas ataupun skripsi.
Pertanyaan 3 : Saya mengalami kesulitan tidur dan sering begadang.
Pertanyaan 4 : Saya mengalami kesulitan keuangan atau keadaan keuangan saya terbatas.
Pertanyaan 5 : Saya merasa kesulitan dalam memahami materi yang disampaikan oleh dosen.
Pertanyaan 6 : Saya merasa kesulitan ketika harus mencari literatur dan referensi untuk tugas atau skripsi.
Pertanyaan 7 : Saya merasa tidak dapat bermain atau istirahat dengan leluasa.
Pertanyaan 8 : Saya merasa tidak nyaman dirumah karena berisik membuat kurang fokus.

Gbr. 2 Pertanyaan pada dataset



Gbr. 3 Alur perancangan model

### E. Pengujian Model

Pengujian ini dilakukan dengan melakukan prediksi terhadap data baru yang di-input. Apabila saat pengujian model tidak dapat mengklasifikasikan tingkat stres mahasiswa dengan akurat, maka penyesuaian/pengaturan manual untuk *hyperparameter* harus dilakukan kembali. Namun, jika model dapat mengklasifikasikan tingkat stres mahasiswa dengan akurat, maka data masukan dari pengguna beserta hasil klasifikasinya dimasukkan ke dalam *dataset* dan disimpan menjadi *dataset* baru dalam bentuk file berformat “csv”. Lalu, *dataset* baru tersebut dilatih kembali untuk menghasilkan model baru. Setelah itu, model baru disimpan ke dalam bentuk file berformat “h5”. Model baru dan *dataset* baru disimpan untuk digunakan diprediksi selanjutnya sehingga dapat dikatakan bahwa model dan *dataset* di dalam penelitian ini bersifat dinamis.



Gbr. 4 Alur pengujian model

## III. HASIL DAN PEMBAHASAN

### A. Hasil Uji Coba Optimalisasi Artificial Neural Network

Dalam proses optimalisasi arsitektur ANN untuk klasifikasi tingkat stres mahasiswa, RMSProp akan dioptimalisasi menggunakan beberapa variasi angka pada tiap *hyperparameter*. *Hyperparameter* yang digunakan pada proses optimalisasi ini adalah *learning rate*, *regularization rate*, jumlah neuron pada *hidden layer* 1 dan 2, dan *epoch*. Selanjutnya, RMSProp diuji performanya menggunakan beberapa metrik evaluasi, yaitu *accuracy*, *precision*, *recall*, *F1-Score*, dan AUC. Pengoptimalan ini akan dilakukan dengan 2

skenario, yaitu pengoptimalan dengan menggunakan *oversampling* SMOTE-N dan tanpa *oversampling* SMOTE-N. Proses optimalisasi ini dilakukan dengan melihat performa tiap variasi angka pada tiap *hyperparameter* dan dilakukan secara bergantian. Pertama, model akan dilihat performanya berdasarkan variasi angka pada *learning rate*. Setelah mendapatkan angka *learning rate* yang menghasilkan performa paling optimal, maka selanjutnya angka *learning rate* tersebut digunakan untuk pengujian performa variasi angka *hyperparameter* selanjutnya, yaitu *regularization rate*. Setelah mendapatkan angka yang paling optimal untuk *regularization rate*, maka selanjutnya angka tersebut digunakan untuk pengujian *hyperparameter* selanjutnya, yaitu *hidden layer* 1. Lalu dilanjutkan dengan pengoptimalan *hidden layer* 2, dan terakhir pengujian pada variasi angka *epoch*. Dari hasil pengujian, akan didapatkan satu percobaan dengan kombinasi *hyperparameter* yang memiliki performa terbaik yang nantinya akan digunakan di dalam model klasifikasi tingkat stres mahasiswa. Dari hasil pengujian juga akan didapatkan pengetahuan, apakah model akan memiliki performa yang lebih baik jika dilakukan *oversampling* atau tidak.

TABEL II

VARIASI ANGKA TIAP HYPERPARAMETER

Jenis Hyperparameter	Variasi Angka
Learning rate	0.0001, 0.001, 0.01, 0.1
Regularization rate	0.0001, 0.001, 0.01, 0.1, 0.2
Hidden layer 1	16, 32, 64, 128, 256, 512
Hidden layer 2	16, 32, 64, 128, 256, 512
Epoch	100, 150, 200, 250, 300

#### 1) Optimalisasi RMSProp dengan SMOTE-N

##### a. Pengujian Learning Rate

Dalam melakukan optimalisasi RMSProp dengan SMOTE-N, *hyperparameter* yang pertama kali diuji performanya adalah *learning rate*. Untuk menguji *learning rate*, model diberi pengaturan awal, yaitu dengan *regularization rate* sebesar 0.1, neuron pada *hidden layer* 1 berjumlah 128 dan *hidden layer* 2 berjumlah 64, serta *epoch* berjumlah 200. Berdasarkan uji coba yang dilakukan, didapatkan kesimpulan bahwa angka *learning rate* yang paling optimal untuk model ini adalah 0.01.

##### b. Pengujian Regularization Rate

Setelah mendapatkan angka *learning rate* yang paling optimal, kemudian RMSProp kembali dioptimalisasi menggunakan angka *learning rate* yang didapatkan pada pengujian sebelumnya dan mengujinya terhadap berbagai variasi angka *regularization rate*. Arsitektur model yang digunakan masih sama seperti pada pengujian *learning rate* sebelumnya, yaitu neuron pada *hidden layer* 1 berjumlah 128, pada *hidden layer* 2 berjumlah 64, dan 200 *epoch*. Dari beberapa variasi angka yang digunakan, didapatkan kesimpulan bahwa angka 0.1 merupakan angka *regularization rate* yang memiliki performa paling optimal.

##### c. Pengujian Hidden Layer 1

Langkah selanjutnya adalah menguji performa RMSProp dengan beberapa variasi jumlah *hidden layer* 1 menggunakan *learning rate* 0.01, *regularization rate* 0.1, 64 neuron pada *hidden layer* 2, dan 200 *epoch*. Didapatkan kesimpulan bahwa

jumlah neuron pada *hidden layer* 1 yang paling optimal adalah 128 neuron.

#### d. Pengujian *Hidden Layer* 2

Setelah didapatkan jumlah neuron pada *hidden layer* 1 yang paling optimal, langkah selanjutnya adalah mencari performa terbaik RMSProp terhadap beberapa variasi jumlah *hidden layer* 2. Dari pengujian ini, didapatkan jumlah neuron pada *hidden layer* 2 yang paling optimal adalah 128 neuron.

#### e. Pengujian *Epoch*

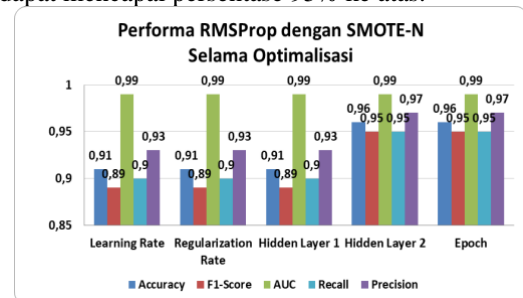
Langkah terakhir pada optimalisasi RMSProp adalah mengujinya dengan beberapa variasi jumlah *epoch* menggunakan *learning rate* 0.01, *regularization rate* 0.1, 128 neuron pada *hidden layer* 1, dan 128 neuron pada *hidden layer* 2. Dari pengujian ini, didapatkan kesimpulan bahwa jumlah *epoch* yang menghasilkan performa paling optimal adalah 200.

TABEL III  
PERFORMA OPTIMALISASI RMSPROP DENGAN SMOTE-N

Hyperparameter	Jumlah	Accuracy	F1-Score	Precision	Recall	AUC
Learning Rate	0.0001	0.77	0.50	0.55	0.62	0.79
	0.001	0.92	0.86	0.87	0.86	0.99
	0.01	0.91	0.89	0.93	0.90	0.99
	0.1	0.29	0.10	0.07	0.33	0.5
	0.0001	0.96	0.55	0.55	0.58	0.81
Regularization Rate	0.001	0.92	0.68	0.71	0.72	0.87
	0.01	0.93	0.74	0.82	0.76	0.98
	0.1	0.91	0.89	0.93	0.90	0.99
	0.2	0.90	0.85	0.86	0.88	0.99
	0.1	0.89	0.85	0.91	0.87	0.98
Hidden Layer 1	16	0.89	0.85	0.91	0.87	0.98
	32	0.89	0.78	0.84	0.8	0.98
	64	0.92	0.86	0.91	0.87	0.98
	128	0.91	0.89	0.93	0.90	0.99
	256	0.90	0.79	0.84	0.81	0.99
Hidden Layer 2	512	0.90	0.78	0.86	0.83	0.97
	16	0.92	0.86	0.91	0.87	0.99
	32	0.90	0.90	0.94	0.91	0.99
	64	0.91	0.89	0.93	0.90	0.99
	128	0.96	0.95	0.97	0.95	0.99
Epoch	256	0.87	0.90	0.94	0.90	0.99
	512	0.91	0.77	0.83	0.79	0.98
	100	0.85	0.77	0.8	0.83	0.97
	150	0.91	0.80	0.85	0.81	0.99
	200	0.96	0.95	0.97	0.95	0.99
	250	0.91	0.94	0.94	0.96	0.99
	300	0.91	0.90	0.93	0.91	0.99

Setelah melakukan optimalisasi RMSProp dengan menggunakan *oversampling* SMOTE-N, didapatkan hasil perolehan performa model seperti pada Gbr. 5. Dari diagram tersebut, dapat disimpulkan bahwa model ini sudah memiliki performa yang cukup baik bahkan sejak awal proses optimalisasi (saat pemilihan angka *learning rate* terbaik). Selanjutnya, performa model mengalami peningkatan yang sangat signifikan di seluruh metrik kecuali AUC saat jumlah *hidden layer* 2 terbaik ditemukan. Metrik *precision* mengalami peningkatan sebesar 4%, *accuracy* dan *recall* sebesar 5%, dan *F1-score* sebanyak 6%. Pada *learning rate*, performa model mulai meningkat signifikan pada semua metrik pada ukuran 0.001 dan mencapai performa maksimum pada ukuran 0.01. Pada *regularization rate*, performa RMSProp dengan SMOTE-N meningkat seiring dengan meningkatnya ukuran

*regularization rate*. Hal ini disebabkan karena *dataset* yang digunakan memiliki sedikit sampel sehingga rawan *overfitting*. Dengan semakin besarnya ukuran *regularization rate*, maka model akan semakin lebih mudah untuk menggeneralisasi data sehingga *overfitting* dapat dihindari. Untuk performa *hidden layer* 1 dan 2 menghasilkan performa yang baik dan mirip di semua variasi jumlah neuron. Hal ini dapat terjadi karena kompleksitas masalah yang dihadapi model tidak begitu tinggi yang menyebabkan perbedaan jumlah variasi neuron tidak memberikan perbedaan performa yang signifikan. Performa RMSProp dengan SMOTE-N menghasilkan peningkatan performa yang signifikan pada *epoch* ke-200. Hasil akhir dari keseluruhan optimalisasi RMSProp dengan SMOTE-N menunjukkan performa yang sangat baik yang mana kelima metrik dapat mencapai persentase 95% ke atas.



Gbr. 5 Performa RMSProp dengan SMOTE-N

## 2) Optimalisasi RMSProp tanpa SMOTE-N

### a. Pengujian *Learning Rate*

Dalam melakukan optimalisasi RMSProp tanpa SMOTE-N, *hyperparameter* yang pertama kali diuji performanya adalah *learning rate*. Untuk menguji *learning rate*, model diberi pengaturan awal, yaitu dengan *regularization rate* sebesar 0.1, neuron pada *hidden layer* 1 berjumlah 128 dan *hidden layer* 2 berjumlah 64, serta *epoch* berjumlah 200. Berdasarkan uji coba yang dilakukan, didapatkan kesimpulan bahwa angka *learning rate* yang paling optimal untuk model ini adalah 0.01.

### b. Pengujian *Regularization Rate*

Setelah mendapatkan angka *learning rate* yang paling optimal, kemudian RMSProp kembali dioptimalisasi menggunakan angka *learning rate* yang didapatkan pada pengujian sebelumnya dan mengujinya terhadap berbagai variasi angka *regularization rate*. Arsitektur model yang digunakan masih sama seperti pada pengujian *learning rate* sebelumnya, yaitu neuron pada *hidden layer* 1 berjumlah 128, pada *hidden layer* 2 berjumlah 64, dan 200 *epoch*. Dari beberapa variasi angka yang digunakan, didapatkan kesimpulan bahwa angka 0.1 merupakan angka *regularization rate* yang memiliki performa paling optimal.

### c. Pengujian *Hidden Layer* 1

Langkah selanjutnya adalah menguji performa RMSProp dengan beberapa variasi jumlah *hidden layer* 1 menggunakan *learning rate* 0.01, *regularization rate* 0.1, 64 neuron pada *hidden layer* 2, dan 200 *epoch*. Didapatkan kesimpulan bahwa jumlah neuron pada *hidden layer* 1 yang paling optimal adalah 256 neuron.



#### d. Pengujian *Hidden Layer 2*

Setelah didapatkan jumlah neuron pada *hidden layer 1* yang paling optimal, langkah selanjutnya adalah mencari performa terbaik RMSProp terhadap beberapa variasi jumlah *hidden layer 2*. Dari pengujian ini, didapatkan jumlah neuron pada *hidden layer 2* yang paling optimal adalah 512 neuron.

#### e. Pengujian *Epoch*

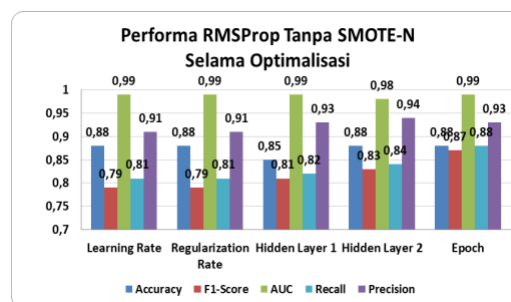
Langkah terakhir pada optimalisasi RMSProp adalah mengujinya dengan beberapa variasi jumlah *epoch* menggunakan *learning rate* 0.01, *regularization rate* 0.1, 256 neuron pada *hidden layer 1*, dan 512 neuron pada *hidden layer 2*. Dari pengujian ini, didapatkan kesimpulan bahwa jumlah *epoch* yang menghasilkan performa paling optimal adalah 300.

TABEL IV  
PERFORMA OPTIMALISASI RMSPROP TANPA SMOTE-N

Hyperparameter	Jumlah	Accuracy	F1-Score	Precision	Recall	AUC
Learning Rate	0.0001	0.66	0.26	0.88	0.33	0.62
	0.001	0.82	0.43	0.91	0.44	0.95
	0.01	0.88	0.79	0.91	0.81	0.99
	0.1	0.66	0.26	0.88	0.33	0.50
Regularization Rate	0.0001	0.94	0.52	0.74	0.53	0.85
	0.001	0.86	0.60	0.83	0.62	0.87
	0.01	0.84	0.75	0.91	0.76	0.96
	0.1	0.88	0.79	0.91	0.81	0.99
	0.2	0.83	0.67	0.93	0.69	0.84
Hidden Layer 1	16	0.89	0.77	0.90	0.79	0.98
	32	0.88	0.76	0.91	0.78	0.99
	64	0.84	0.75	0.90	0.76	0.97
	128	0.88	0.79	0.91	0.81	0.99
	256	0.85	0.81	0.93	0.82	0.99
	512	0.84	0.76	0.91	0.78	0.98
	1024	0.86	0.79	0.92	0.79	0.98
Hidden Layer 2	16	0.86	0.79	0.92	0.79	0.98
	32	0.86	0.80	0.92	0.81	0.98
	64	0.85	0.81	0.93	0.82	0.99
	128	0.89	0.80	0.93	0.81	0.99
	256	0.83	0.81	0.94	0.81	0.98
	512	0.88	0.83	0.94	0.84	0.98
	1024	0.80	0.47	0.84	0.55	0.89
Epoch	100	0.80	0.47	0.84	0.55	0.89
	150	0.78	0.59	0.87	0.63	0.97
	200	0.88	0.83	0.94	0.84	0.98
	250	0.89	0.83	0.92	0.83	0.99
	300	0.88	0.87	0.93	0.88	0.99

Setelah melakukan optimalisasi RMSProp tanpa menggunakan SMOTE-N, didapatkan hasil perolehan performa model seperti pada Gbr. 6. Saat proses pengoptimalan dengan pemilihan jumlah neuron terbaik pada *hidden layer 1*, terjadi peningkatan di metrik *F1-score* dan *precision* sebesar 2%, serta *recall* sebesar 1%. Namun di saat yang sama, model juga mengalami penurunan *accuracy* sebesar 3%. Selanjutnya pada pemilihan jumlah neuron terbaik pada *hidden layer 2*, model mengalami peningkatan *F1-score* dan *recall* sebesar 2%, *accuracy* sebesar 3%, dan *precision* sebesar 1%. Namun di saat yang sama, model juga mengalami penurunan AUC sebesar 1%. Terakhir, pada pemilihan jumlah *epoch* terbaik, terjadi peningkatan *F1-score* sebesar 4%, *recall* 3%, dan AUC 1%. Namun di saat yang sama, model juga mengalami penurunan *precision* sebesar 1%. Pada *learning rate*, performa model meningkat signifikan pada semua metrik pada ukuran 0.01. Pada *regularization rate*, performa RMSProp tanpa SMOTE-N meningkat seiring dengan meningkatnya ukuran *regularization rate*. Hal ini disebabkan karena *dataset* yang digunakan

memiliki sedikit sampel sehingga rawan *overfitting*. Dengan semakin besarnya ukuran *regularization rate*, maka model akan semakin lebih mudah untuk menggeneralisasi data sehingga *overfitting* dapat dihindari. Namun, generalisasi yang berlebihan pada ukuran *regularization rate* 0.2 menyebabkan model mengalami penurunan performa. Untuk performa *hidden layer 1* dan *hidden layer 2* menghasilkan performa yang cukup baik dan mirip di semua variasi jumlah neuron. Hal ini dapat terjadi karena kompleksitas masalah yang dihadapi model tidak begitu tinggi yang menyebabkan perbedaan jumlah variasi neuron tidak memberikan perbedaan performa yang signifikan. Performa RMSProp tanpa SMOTE-N menghasilkan peningkatan performa pada *epoch* ke-200 mencapai performa maksimum pada *epoch* ke-300. Hal ini menandakan bahwa RMSProp tanpa SMOTE-N masih bisa mencapai kinerja baik (87% *F1-Score* pada *epoch* ke-300), walaupun lebih lambat dibanding RMSProp dengan SMOTE-N yang dapat mencapai kinerja sangat baik (95% *F1-Score* pada *epoch* ke-200). Hasil akhir dari keseluruhan optimalisasi RMSProp tanpa SMOTE-N menunjukkan performa yang cukup baik yang mana kelima metrik dapat mencapai persentase 87% ke atas.



Gbr. 6 Performa RMSProp tanpa SMOTE-N

#### B. Pengujian Model untuk Klasifikasi Stres Mahasiswa

Pada tahap optimalisasi model, telah didapatkan hasil model terbaik, yaitu RMSProp dengan SMOTE-N. Setelah didapatkan satu model terbaik, maka langkah selanjutnya adalah melakukan pengujian model. Pertama-tama, pengguna akan diberikan 8 pertanyaan mengenai kehidupan pribadi pengguna dan mereka dapat mengisinya dengan 4 range skor, yaitu 1 untuk sangat tidak setuju, 2 untuk tidak setuju, 3 untuk setuju, dan 4 untuk sangat setuju. Selanjutnya, model akan mengklasifikasikan tingkat stres berdasarkan masukan pengguna dan mengeluarkan hasil prediksi. Berdasarkan hasil pengujian yang telah dilakukan, model yang telah dioptimalisasi berhasil melakukan klasifikasi tingkat stres dengan sangat baik.

Saya merasa tertekan oleh tuntutan orang tua yang menyuruh saya segera wisuda.  
Jawaban: 1  
Teman saya selalu mengajak saya bermain, sehingga saya merasa tidak fokus dalam mengerjakan tugas  
Jawaban: 1  
Saya mengalami kesulitan tidur dan sering begadang.  
Jawaban: 1  
Saya mengalami kesulitan keuangan atau keadaan keuangan saya terbatas.  
Jawaban: 1  
Saya merasa kesulitan dalam memahami materi yang disampaikan oleh dosen.  
Jawaban: 1  
Saya merasa kesulitan ketika harus mencari literatur dan referensi untuk tugas atau skripsi.  
Jawaban: 1  
Saya merasa tidak dapat bermain atau istirahat dengan leluasa.  
Jawaban: 1  
Saya merasa tidak nyaman di rumah karena berisik membuat kurang fokus.  
Jawaban: 1  
1/1 [=====] - 0s 53ms/step  
Hasil prediksi tingkat stres: RINGAN

Gbr. 7 Prediksi tingkat stres ringan

Saya merasa tertekan oleh tuntutan orang tua yang menyuruh saya segera wisuda.  
Jawaban: 3  
Teman saya selalu mengajak saya bermain, sehingga saya merasa tidak fokus dalam mengerjakan tugas  
Jawaban: 2  
Saya mengalami kesulitan tidur dan sering begadang.  
Jawaban: 3  
Saya mengalami kesulitan keuangan atau keadaan keuangan saya terbatas.  
Jawaban: 2  
Saya merasa kesulitan dalam memahami materi yang disampaikan oleh dosen.  
Jawaban: 3  
Saya merasa kesulitan ketika harus mencari literatur dan referensi untuk tugas atau skripsi.  
Jawaban: 2  
Saya merasa tidak dapat bermain atau istirahat dengan leluasa.  
Jawaban: 3  
Saya merasa tidak nyaman di rumah karena berisik membuat kurang fokus.  
Jawaban: 2  
1/1 [=====] - 0s 51ms/step  
Hasil prediksi tingkat stres: SEDANG

Gbr. 8 Prediksi tingkat stres sedang

Saya merasa tertekan oleh tuntutan orang tua yang menyuruh saya segera wisuda.  
Jawaban: 4  
Teman saya selalu mengajak saya bermain, sehingga saya merasa tidak fokus dalam mengerjakan tugas  
Jawaban: 4  
Saya mengalami kesulitan tidur dan sering begadang.  
Jawaban: 4  
Saya mengalami kesulitan keuangan atau keadaan keuangan saya terbatas.  
Jawaban: 4  
Saya merasa kesulitan dalam memahami materi yang disampaikan oleh dosen.  
Jawaban: 4  
Saya merasa kesulitan ketika harus mencari literatur dan referensi untuk tugas atau skripsi.  
Jawaban: 4  
Saya merasa tidak dapat bermain atau istirahat dengan leluasa.  
Jawaban: 4  
Saya merasa tidak nyaman di rumah karena berisik membuat kurang fokus.  
Jawaban: 4  
1/1 [=====] - 0s 51ms/step  
Hasil prediksi tingkat stres: BERAT

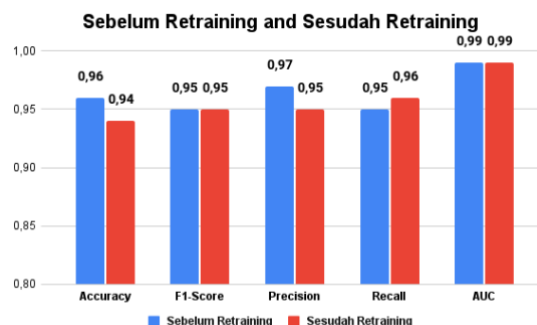
Gbr. 9 Prediksi tingkat stres berat

Karena model telah menunjukkan hasil prediksi yang baik, maka langkah selanjutnya adalah menguji ketahanan model terhadap sampel data baru. Dalam pengujian ini, setiap *input* pengguna akan dimasukkan ke dalam *dataset* dan disimpan menjadi *dataset* baru dalam format “csv”. Pada penelitian ini, dimasukkan ke dalam *dataset* sebanyak 50 sampel data tambahan yang merupakan hasil prediksi model yang sebelumnya dibuat. Setelah itu, model di-*training* ulang menggunakan *dataset* yang telah diperbarui dan model yang telah di-*retraining* disimpan ke dalam format “h5” untuk digunakan pada prediksi selanjutnya. Hasil performa model yang telah di-*retraining* menggunakan *dataset* baru menunjukkan performa yang tidak berbeda jauh dengan model sebelum di-*retraining*. Seperti yang terlihat pada Gbr. 11, pada metrik *accuracy* dan *precision* hanya mengalami penurunan sebesar 2%, *F1-score* dan AUC stagnan, serta *recall* mengalami kenaikan sebesar 1%. Hal ini menandakan bahwa model yang telah dibuat memiliki ketahanan yang baik dalam memprediksi data baru yang belum diketahui.

Dataset setelah penambahan data baru:

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Hasil	Tingkat	Stres
0	2	2	3	3	3	3	2	2			SEDANG
1	4	2	4	4	4	4	3	3			BERAT
2	2	2	2	2	3	3	3	3			SEDANG
3	2	2	3	4	3	3	3	3			SEDANG
4	4	2	3	3	3	3	2	3			SEDANG
..	..	..	..	..	..	..	..	..			...
145	4	2	4	2	4	2	4	2			BERAT
146	1	1	1	1	1	2	2	2			RINGAN
147	2	3	3	3	4	4	1	1			SEDANG
148	1	1	1	2	2	2	2	3			RINGAN
149	3	3	3	4	4	1	1	1			SEDANG

Gbr. 10 Dataset setelah ditambahkan masukan pengguna



Gbr. 11 Perbandingan hasil performa model sebelum dan sesudah *retraining*

#### IV. KESIMPULAN

Keseluruhan penelitian mengenai metode *Artificial Neural Network* (ANN) untuk klasifikasi tingkat stres mahasiswa berhasil dilakukan. Adapun kesimpulan dari penelitian ini adalah sebagai berikut:

- 1) Proses pengoptimalan ANN berhasil terhadap *optimizer* RMSProp dengan berbagai pengaturan *hyperparameter* dan dengan percobaan menggunakan dan tanpa menggunakan *oversampling* SMOTE-N. Dari pengujian yang telah dilakukan, didapatkan kesimpulan bahwa teknik *oversampling* data menggunakan SMOTE-N dapat meningkatkan performa model di hampir semua metrik, terutama *recall* dan *F1-score*. Hal ini disebabkan karena *recall* dan *F1-score* bersifat kritis terhadap kelas minoritas sehingga ketidakseimbangan data yang digunakan pada optimalisasi tanpa SMOTE-N menyebabkan nilai *recall* dan *F1-score* menjadi rendah.
- 2) Model RMSProp dengan SMOTE-N diuji dengan melakukan prediksi pada input baru yang diberikan oleh pengguna. Hasil prediksi menunjukkan bahwa model dapat mengklasifikasikan tingkat stres dengan sangat baik, baik itu di tingkat stres ringan, sedang, maupun berat. Selain itu, ketika model di-*retraining* dengan *dataset* baru yang memiliki 50 sampel data tambahan dibanding *dataset* asli, didapatkan hasil performa model setelah *retraining* yang tidak jauh berbeda dengan performa model sebelum *retraining*. Hal ini menandakan bahwa model yang digunakan dapat mengklasifikasikan tingkat stres mahasiswa dengan akurat dan memiliki ketahanan yang baik terhadap data baru.

## V. SARAN

Penelitian klasifikasi tingkat stres mahasiswa ini belum pada tahap pembuatan sistem yang memiliki GUI (*Graphic User Interface*). Hal ini menyebabkan hasil penelitian belum dapat digunakan untuk masyarakat luas. Harapannya, penelitian selanjutnya yang berkaitan dengan klasifikasi tingkat stres dapat menciptakan sistem yang memiliki GUI dan dapat diakses oleh masyarakat luas, khususnya mahasiswa. Selain itu, perlu juga adanya edukasi penanganan stres sesuai dengan tingkat stres di dalam sistem. Hal ini bertujuan agar masyarakat dapat mengetahui tingkat stres yang dialaminya dan mendapat edukasi penanganan stres yang tepat agar terhindar dari berbagai macam dampak dari stres berkepanjangan.

## UCAPAN TERIMA KASIH

Rasa terima kasih dengan tulus penulis ucapkan kepada Allah yang telah memberi karunia dan rahmat-Nya sehingga artikel ilmiah ini dapat terselesaikan dengan baik. Terima kasih juga penulis sampaikan kepada beberapa pihak, seperti orang tua, dosen pembimbing, keluarga, kerabat, dan teman-teman kuliah yang telah memberikan dukungan serta doa kepada penulis.

## REFERENSI

- [1] S. Widjaja and A. Cahyono, "Stress Analytics of Medical Students At Faculty of Medicine University of Surabaya," *J. Pendidikan Kedokteran Indonesia: Indonesian J. Med. Educ.*, vol. 11, no. 3, p. 319, 2022. [Online]. Available: <https://doi.org/10.22146/jpki.64881>
- [2] S. B. Seto, M. T. S. Wondo, and M. F. Mei, "Hubungan Motivasi Terhadap Tingkat Stress Mahasiswa Dalam Menulis Tugas Akhir (Skripsi)," *J. Basicedu*, vol. 4, no. 3, pp. 733–739, 2020. [Online]. Available: <https://doi.org/10.31004/basicedu.v4i3.431>
- [3] F. Handayani and H. Fithroni, "Hubungan Tingkat Aktivitas Fisik terhadap Tingkat Stres Mahasiswa Akhir Fakultas Ilmu Olahraga Universitas Negeri Surabaya," *J. Kesehat. Olahraga*, vol. 10, no. 03, pp. 131–138, 2022.
- [4] Merry and H. C. Mamahit, "Stres Akademik Mahasiswa Aktif Angkatan 2018 dan 2019 Universitas Swasta di DKI Jakarta," *J. Konseling Indonesia*, vol. 6, no. 1, pp. 6–13, 2020.
- [5] F. U. N. Erindana, H. F. Nashori, and M. N. F. Tasaufi, "Penyesuaian Diri Dan Stres Akademik Mahasiswa Tahun Pertama Self Adjustment and Academic Stress in First-Year University Student," *Motiva: J. Psikol.*, vol. 4, no. 1, pp. 11–18, 2021.
- [6] D. D. W. Sari and W. M. Wulan, "Klasifikasi Tingkat Stres Akademik dan Gambaran Mekanisme Koping Mahasiswa," pp. 203–212, 2023.
- [7] M. Wibowo and M. R. F. Djafar, "Perbandingan Metode Klasifikasi Untuk Deteksi Stress Pada Mahasiswa di Perguruan Tinggi," *J. Media Informatika Budidarma*, vol. 7, no. 1, pp. 153–159, 2023. [Online]. Available: <https://doi.org/10.30865/mib.v7i1.5182>
- [8] A. Anjani and Y. Yamasari, "Klasifikasi Tingkat Stres Mahasiswa Menggunakan Metode Berbasis Tree," *J. Informatics Comput. Sci.*, vol. 05, pp. 83–89, 2023.
- [9] R. S. Nancy Noella and J. Priyadarshini, "Diagnosis of Alzheimer's and Parkinson's Disease Using Artificial Neural Network," *Int. J. Scientific Technol. Res.*, vol. 9, no. 3, pp. 3659–3664, 2020.
- [10] L. Istianah and H. Sumarti, "Classification of Pneumonia in Thoracic X-Ray images based on texture characteristics using the MLP (Multi-Layer Perceptron) method," *J. Nat. Sci. Math. Res.*, vol. 6, no. 2, pp. 78–84, 2020.
- [11] Z. Car, S. Baressi Šegota, N. Anđelić, I. Lorencin, and V. Mrzljak, "Modeling the Spread of COVID-19 Infection Using a Multilayer Perceptron," *Comput. Math. Methods Med.*, 2020.
- [12] S. Samarpita, R. Satpathy, P. Kumar Mishra, and A. Narayan Panda, "Mental Stress Classification from Brain Signals using MLP Classifier," *EAI Endorsed Trans. Pervasive Health Technol.*, vol. 9, pp. 1–6, 2023.
- [13] C. Desai, "Comparative Analysis of Optimizers in Deep Neural Networks," *Int. J. Innovative Sci. Res. Technol.*, vol. 5, no. 10, pp. 959–962, 2020.
- [14] R. Maya and F. Lubis, "Pengembangan Analisa Algoritma Autoregressive Integrated Moving Average ( Arima- Box Jenkins ) Pemodelan Menggunakan Google Colab ( Python )," pp. 44–56, 2019.
- [15] A. N. Syahrudin and T. Kurniawan, "Input Dan Output Pada Bahasa Pemrograman Python," *Jurnal Dasar Pemograman Python STMIK*, 2018.