Klasifikasi Kualitas Udara Menggunakan Metode *Extreme Learning Machine* (Studi Kasus: ISPU DKI Jakarta)

1st Wahyu Mubarak Sukiman
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
wahyumubarak@student.telkomuniversity.ac.id

2nd Meta Kallista
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
metakallista@telkomuniversity.ac.id

3rd Ig. Prasetya Dwi Wibawa
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
prasdwibawa@telkomuniversity.ac.id

***Abstrak* — Kualitas udara yang baik sangat berpengaruh untuk menjaga keberlangsungan kehidupan. Kualitas udara sangat berpengaruh kepada kualitas oksigen yang dibutuhkan oleh tubuh manusia. Polusi udara adalah salah satu komponen yang sangat mempengaruhi kualitas oksigen. Ibu kota Indonesia, Jakarta, menduduki peringkat ke-9 untuk kualitas udara dan polusi perkotaan. Informasi kualitas udara tentunya sangat dibutuhkan manusia. Masyarakat perlu mengetahui informasi tentang kualitas udara agar lebih peduli terhadap pengaruh polusi udara terhadap kesehatannya. Informasi kualitas udara yang dibutuhkan adalah indeks kualitas udara. Oleh karena pada penelitian ini dilakukan klasifikasi terhadap indeks kualitas udara. Proses klasifikasi dilakukan menggunakan algoritma *machine learning* dengan metode *Extreme Learning Machine* (ELM). Algoritma *Extreme Learning Machine* (ELM) dipilih karena memiliki kelebihan pembelajaran lebih cepat, mudah digunakan untuk masalah kompleks, dan relevan dengan dunia nyata. Dataset yang digunakan untuk penelitian ini berasal dari Jakarta Open Data dan Jakarta Rendah Emisi. Penelitian ini membuktikan bahwa penggunaan *machine learning* dengan metode *Extreme Learning Machine* (ELM) efektif dalam melakukan proses klasifikasi. Dalam proses klasifikasi, *Extreme Learning Machine* (ELM) menghasilkan performa baik dengan menggunakan data *balance* maupun data *imbalance*. Percobaan dengan data *imbalance* dengan akurasi tinggi sebesar 94% dan data yang *balance* dengan akurasi sebesar 96%.**

***Kata kunci*— extreme learning machine, kualitas udara, klasifikasi, machine learning**.

I. PENDAHULUAN

Menurut data yang diperoleh dari IQAir, pada bulan Oktober 2021, Ibu kota Indonesia, Jakarta, menduduki peringkat ke-9 untuk kualitas udara dan polusi kota. Dari 106 negara, Indonesia menempati peringkat ke-9 untuk negara paling berpolusi pada tahun 2020 dilihat dari konsentrasi partikel PM2.5 [1]. Sehingga, penelitian ini dilakukan untuk melakukan klasifikasi kualitas udara sesuai dengan Indeks Standar Pencemaran Udara (ISPU).

DKI Jakarta merupakan kota yang memiliki kepadatan kendaraan bermotor yang sangat tinggi. Sekitar 70% polutan berasal dari emisi kendaraan. Kendaraan bermotor adalah salah satu penyebab polusi terbesar, dikarenakan banyak mengeluarkan zat yang sangat berbahaya bagi kesehatan manusia dan merusak lingkungan[2].

Penelitian sebelumnya melakukan klasifikasi kualitas udara menggunakan algoritma Support Vector Machine (SVM), tetapi penelitian ini melakukan proses yang sama dengan metode yang berbeda. Extreme Learning Machine (ELM) adalah metode yang akan digunakan.

Metode *Extreme Learning Machine* (ELM) digunakan karena metode ini memiliki kelebihan pembelajaran lebih cepat, mudah digunakan untuk masalah kompleks dan relevan dengan kehidupan nyata[3]. Penelitian sebelumnya telah banyak melakukan penelitian mengenai metode Extreme Learning Machine (ELM). Salah satu penelitian tersebut adalah penelitian oleh Prakoso, Wisesty & Jondri (2016) dengan penelitian yang berjudul Klasifikasi Keadaan Mata Berdasarkan Sinyal EEG Menggunakan *Extreme Learning Machine*. Penelitian ini memiliki nilai akurasi sebesar 97,95%, yang membuatnya sangat baik.

Salah satu faktor yang menyebabkan hasil klasifikasi kurang maksimal yaitu data yang tidak seimbang. Untuk memecahkan permasalahan tersebut perlu dilakukan proses untuk menyeimbangkan data minoritas dengan data mayoritas. *Synthetic Minority Over-Sampling Technique* (SMOTE) adalah salah satu metode yang digunakan untuk menyeimbangkan data. Untuk mendapatkan hasil klasifikasi yang baik, proses menyeimbangkan data cukup berpengaruh. Selain itu, penelitian ini juga membandingkan hasil klasifikasi menggunakan metode *Extreme Learning Machine* (ELM) yang menggunakan data yang tidak seimbang dan data yang seimbang,

II. KAJIAN TEORI

A. Kualitas Udara

Kualitas udara didefinisikan sebagai ukuran seberapa baik atau buruk campuran gas pada lapisan toposfer yang diperlukan. Kualitas udara dapat mempengaruhi kesehatan manusia, makhluk hidup, dan elemen lingkungan[4].

B. Pencemaran Udara

Menurut Undang-Undang Pokok Pengolahan Lingkungan Hidup No. 4 Tahun 1982, Pencemaran udara terjadi ketika makhluk hidup, zat energi, dan atau komponen lain masuk ke dalam lingkungan, atau ketika struktur lingkungan diubah oleh aktivitas manusia atau alam itu sendiri sehingga kualitas lingkungan menjadi buruk atau tidak dapat berfungsi dengan baik.

Terdapat berbagai macam senyawa jika bercampur dengan udara dapat menyebabkan udara menjadi tercemar, senyawa ini termasuk *Karbon Monoksida* (CO), *Sulfur Dioksida* (SO2), *Nitrogen Dioksida* (NO2), *Ozon* (O3), dan *Particulate 10 Micron* (PM10)[5]. Kualitas udara yang buruk dapat berdampak langsung pada kesehatan manusia.

C. Pengambilan Data

Data yang digunakan pada penelitian ini merupakan dataset Indeks Standar Pencemaran Udara (ISPU) pada 5 titik di DKI Jakarta. Kelima lokasi tersebut yaitu DKI 1 Bunderan HI, DKI 2 Kelapa Gading, DKI 3 Jagakarsa, DKI 4 Lubang Buaya, DKI 5 Kebon Jeruk. Pada penilitian ini, dilakukan klasifikasi dari partikel yang ada pada data ISPU untuk menentukan indeks kualitas udaranya. Partikel yang terdapat pada dataset ini, *Particulate 10 Micron* (PM_{10}), *Sulfur Dioxide* (SO_2), *Carbon Monoxide* (CO), Ozon (O_3), *Nitrogen Dioxide* (NO_2).

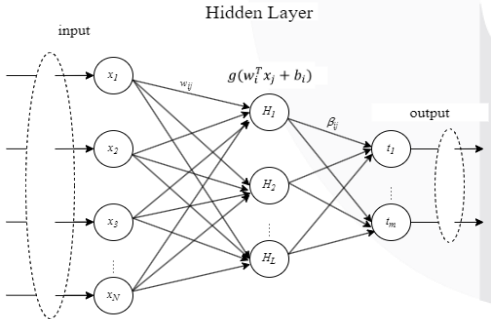
D. Klasifikasi

Klasifkasi adalah suatu proses untuk menemukan nilai fungsi yang menjelaskan suatu kelas data. Tujuan dari klasifikasi ini adalah untuk memperkirakan kelas suatu objek atau label yang nilainya belum diketahui sebelumnya[6].

E. *Extreme Learning Machine* (ELM)

Arsitektur Extreme Learning Machine (ELM), yang dipopulerkan oleh Guang-Bin Huang, terdiri dari satu lapisan input, satu lapisan tersembunyi, dan satu lapisan output. ELM belajar lebih cepat daripada algoritma lainnya dan memiliki generalisasi yang baik, sehingga kemungkinan errornya kecil. Bobot maupun bias metode ELM ditentukan secara acak dan dikerjakan secara bersamaan. Hal ini akan menghasilkan hasil prediksi dan mempercepat proses pembelajaran.

Pada gambar 1 berikut adalah arsitektur dari metode ELM:



GAMBAR 1
Arsitektur *ELM*

Proses menghitung pada klasifikasi menggunakan metode ELM terbagi jadi 2, diantaranya proses *training* dan *testing* yang dijelaskan pada tahapan-tahapan pada klasifikasi menggunakan metode ELM.

F. *Confusion Matrix*

Tujuan dari *Confusion Matrix* adalah untuk mengevaluasi hasil proses metode klasifikasi biner atau *multi-class* dengan membandingkan hasil klasifikasi dari prose *machine learning* dengan hasil dari klasifikasi data asli. Fungsi dari *Confusion Matrix* adalah untuk mengetahui seberapa baik metode klasifikasi biner atau multikelas bekerja. Ini dilakukan dengan membandingkan hasil klasifikasi dari *machine learning* dengan hasil klasifikasi data asli. Parameter *sensitivity*, *specificity*, *g-mean*, dan *accuracy* digunakan dalam proses evaluasi. Tabel hasil *confusion matrix* bisa dilihat pada tabel 1 berikut.

TABEL 1
Tabel hasil *confusion matrix*

Label Aktual	**Label Prediksi**	
	Positive	Negative
Positif	True Positive	False Negative
Negative	False Positive	True Negative

Berikut penjelasan istilah pada tabel 1 untuk menghitung hasil klasifikasi:

1. True Positive (TP) yaitu ketika sistem berhasil mengklasifikasikan dengan benar data yang bersifat positif.
2. True Negative (TN) yaitu ketika sistem berhasil mengklasifikasikan dengan benar data yang bersifat negatif.
3. False Positive (FP) yaitu ketika sistem mengklasifikasikan data bersifat negatif menjadi data positif.
4. False Negative (FN) yaitu ketika sistem mengklasifikasikan data bersifat positif menjadi data negatif.

Sensitivity mengukur kemampuan model untuk mengidentifikasi dengan benar semua instance yang positif dalam kelas yang relevan. Perhitungan *sensitivity* menggunakan persamaan (1)

$$Sensitivity = \frac{TP}{(TP + FN)} \quad (1)$$

Spesificity mengukur kemampuan model untuk mengidentifikasi dengan benar semua instance yang negatif dalam kelas yang relevan. Perhitungan *spesificity* menggunakan persamaan (2)

$$Specificity = \frac{TN}{(TN + FP)} \quad (2)$$

G-mean adalah akar kuadrat dari hasil kali sensitivitas dan spesifisitas. Metrik ini membantu untuk menemukan keseimbangan antara sensitivitas dan spesifisitas dalam suatu model klasifikasi. Perhitungan *g-mean* menggunakan persamaan (3)

$$G - mean = \sqrt{(Sensitivity * Specificity} \quad (3)$$

Accuracy mengukur kebenaran secara keseluruhan dari prediksi model. Perhitungan *accuracy* menggunakan persamaan (4)

$$Accuracy = \frac{(Jumlah\ Prediksi\ Benar)}{(Jumlah\ Total\ Data)} \quad (4)$$

G. *Synthetic Minority Over-Sampling Technique (SMOTE)*

Pada proses klasifikasi, akurasi merupakan hal yang sangat penting. Semakin tinggi nilai akurasi ,semakin baik juga *output* yang dihasilkan oleh proses klasifikasi. Salah satu penyebab akurasi dari proses klasifikasi tidak maksimal yaitu pada data yang digunakan. Contohnya jika menggunakan data yang tidak seimbang akurasi yang didapatkan kurang maksimal.

Untuk mengatasi hal tersebut yang perlu dilakukan yaitu membuat data yang tidak seimbang menjadi seimbang. Salah satu cara untuk menyeimbangkan data yaitu *oversampling* pada kelas minoritas atau *undersampling* pada kelas mayoritas[7]. Penelitian ini menggunakan teknik *oversampling* untuk menyeimbangkan data. Teknik yang digunakan yaitu *Synthetic Minority Over-Sampling* Technique (SMOTE).

H. Matriks Evaluasi ROC-AUC

Untuk mengukur kinerja dari proses klasifikasi salah satu matriks evaluasi juga yang dipakai yaitu ROC-AUC. ROC-AUC adalah *Receiver Operating Characteristic Area Under the Curve*. Matriks evaluasi ini fokus untuk membandingkan antara kelas positif dan juga kelas negatif dengan menampilkan trade-off antara *True Positive Rate* dan *False Positive Rate* [8].

III. METODE

Jaringan syaraf tiruan atau single hidden layer feedforward neural network (SLNs) termasuk Extreme Learning Machine (ELM). ELM mempunyai *learning speed* lebih cepat jika dibandingkan dengan beberapa metode *feedforward*, seperti *Backpropagation* dan performa generalisasi lebih baik [9]. Pada JST, semua parameter, termasuk input dan bias, harus ditentukan, sehingga menghabiskan waktu lebih lama untuk belajar. Tetapi, pada ELM, parameter yang digunakan dipilih secara acak, sehingga waktu yang digunakan lebih sedikit untuk belajar serta mampu menghasilkan hasil yang baik[10].

Tahapan – tahapan pada klasifikasi menggunakan metode ELM dijelaskan pada sub bab dibawah:

A. *Preprocessing* dan *Balancing* Data

Sebelum lanjut pada proses klasifikasi, perlu dilakukan proses *preprocessing*. *Preprocessing* data bertujuan untuk mempersiapkan dan memperbaiki data agar proses klasifikasi berjalan dengan baik dan lancar.

Setelah melakukan *preprocessing* data, perlu dilakukan *balancing* data. *Balancing* data bertujuan untuk menyeimbangkan nilai *class* pada dataset yang digunakan, sehingga melancarkan proses klasifikasi serta mendapatkan hasil klasifikasi yang baik. *Balancing* data juga bertujuan untuk membandingkan hasil dari proses klasifikasi dengan menggunakan data *imbalance* dan data *balance*. Teknik yang dilakukan pada proses *balancing* data yaitu *Synthetic Minority Over-Sampling Technique* (SMOTE).

B. Proses *Training*

Tahap 1:

Menentukan nilai (W) dan bias secara acak pada matriks berukuran h x d, h merupakan jumlah *hidden neuron* serta k yaitu banyak *input node*.

Tahap 2:

Melakukan perhitungan pada matriks keluaran lapisan tersembunyi (H) serta menggunakan fungsi aktivasi sigmoid untuk menghasilkan matriks keluaran lapisan tersembunyi pada rentang 0 hingga 1 dengan persamaan (5)

$$H = 1/(1 + \mathrm{e}\ (-(X.W^T + ones(N_{train}, 1)b))) \quad (5)$$

H = Keluaran lapisan tersembunyi
e = Eksponensial
X = Matriks berukuran data masukan
W^T = Matriks *Transpose*
$ones(N_{train},1)$ = matriks yang memiliki nilai 1 dengan jumlah baris sama dengan jumlah pada data *training*.
b = Bias

Tahap 3:

Matriks *Moore-Penrose Pseudo Inverse* dihitung dengan perkalian matriks *inverse* dan *transpose* keluaran lapisan tersembunyi. Perhitungan matriks ini ditunjukkan pada persamaan. persamaan (6)

$$H^+ = (H^T.H)^{-1}.H^T \quad (6)$$

H^+ = Matriks Moore–Penrose Pseudo Inverse
H = Matriks keluaran lapisan tersembunyi
H^T = Transpose H

Tahap 4:

Melakukan penghitungan output weight (β) yang dihasilkan dari lapisan tersembunyi dan lapisan keluaran menggunakan persamaan (7)

$$\beta = H^+.Y \quad (7)$$

β = Output weight
H^+ = Matriks Moore–Penrose Pseudo Inverse
Y = Matriks keluaran

Tahap 5:

Menghitung hasil prediksi (Y) dari proses perkalian antara matriks keluaran lapisan tersembunyi dan *output weight* menggunakan persamaan (8)

$$Y = H.\beta \quad (8)$$

Keterangan:
Y = Hasil Prediksi
H = Matriks Keluaran lapisan tersembunyi
β = Matrik Output weight

C. Proses *Testing*

Sama dengan proses *training*. Tetapi proses *testing* tidak perlu menghitung input *weight* , bias, dan output *weight*. sebaliknya, tahap ini menggunakan bobot yang telah dihitung selama proses pelatihan.

Tahap 1:

Dari proses *training* bobot dan bias diperoleh secara acak.

Tahap 2:

Menghitung matriks H menggunakan persamaan (9)

$$H = 1/(1 + \mathrm{e}\ (-(X.W^T + ones(N_{test}, 1)b))) \quad (9)$$

H = Keluaran lapisan tersembunyi
e = Eksponensial
X = Matriks data masukan
W^T = Matriks *Transpose weight*
$ones(N_{test},1)$ = matriks bernilai 1 dengan jumlah baris sama dengan jumlah pada data *training* dan 1 kolom.

b = Bias
Tahap 3:
Perhitungan hasil prediksi (Y) menggunakan persamaan (10)

$$Y = H.\beta \quad (10)$$

Y = Hasil Prediksi
H = Matriks keluaran lapisan tersembunyi
β = Matrik *Output weight*

D. Parameter

Pada klasifikasi menggunakan ELM, terdapat beberapa parameter yang digunakan. Berikut ini adalah parameter yang digunakan pada proses klasifikasi menggunakan metode ELM.

Fungsi aktivasi adalah fungsi matematika yang diterapkan pada keluaran neuron dalam jaringan saraf. Fitur ini memperkenalkan *non-linearitas* ke dalam jaringan saraf dan memungkinkan model mempelajari hubungan kompleks antara karakteristik *input* dan *output* yang diinginkan.

Pada penelitian ini terdapat 3 fungsi aktivasi yaitu *Sigmoid*, *Sin*, dan *ReLU*.

a) Sigmoid

Fungsi aktivasi sigmoid untuk mengonversi nilai input menjadi rentang (0, 1), menghasilkan *output* yang terbatas di antara dua nilai.
Berikut rumus fungsi aktivasi Sigmoid:

$$f(\mathrm{x}) = 1/(1 + \exp(-\mathrm{x})) \quad (11)$$

b) Sin (Sinus)

Fungsi aktivasi sinus digunakan untuk memberikan output dalam rentang [-1, 1], sehingga menghasilkan fungsi non-linear simetris.
Berikut rumus fungsi aktivasi Sin:

$$f(\mathrm{x}) = \sin(x) \quad (12)$$

c) ReLU (*Rectified Linear Unit*)

Fungsi aktivasi ReLU mengaktifkan neuron hanya ketika nilai inputnya positif, yaitu *output* sama dengan nilai inputnya jika *input* lebih besar dari nol, dan output nol jika input kurang dari atau sama dengan nol.
Berikut rumus fungsi aktivasi ReLU:

$$f(x) = \max(0, x) \quad (13)$$

Regularisasi Parameter atau C merupakan paremeter yang bertujuan untuk mengatasi *overfitting* dan mengontrol kompleksitas model lebih lanjut. Angka yang digunakan yaitu 10, 100, 1000.

Jumlah *hidden neuron* merupakan parameter yang sangat penting pada model jaringan saraf, salah satunya pada model ELM (*Extreme Learning Machine*). Jumlah neuron tersembunyi menentukan kompleksitas dan kapasitas model, serta kemampuan model untuk belajar dari data pelatihan dan mempelajari pola. Jumlah hidden neuron yang digunakan yaitu 100, 300, 500.

Random Type, parameter yang dapat mengatur atau menentukan inisialisasi bobot yang terhubung antara data *input* dan *hidden layer*.

E. Evaluasi

Untuk mengetahui seberapa baik hasil klasifikasi yang dibuat, perlu dilakukan proses evaluasi. Pada penelitian ini, evaluasi dilakukan dengan menganalisis hasil dari *confusion matrix* dan menghitung matriks evaluasi yaitu *sensitivity*, *specificity*, *g-mean*, *accuracy*.

IV. HASIL DAN PEMBAHASAN

Pada penelitian ini memiliki hasil akhir berupa klasifikasi menggunakan dataset *imbalance* dan *balance*. Pada algoritma klasifikasi menggunakan metode ELM digunakan beberapa parameter untuk mendapatkan hasil terbaik, yaitu *split data*, fungsi aktivasi, nilai C pada regularisasi parameter, *random type*, dan jumlah *hidden neuron*. Split data yang diuji adalah 70%:30% dan 80%:20%, untuk nilai C yaitu 10, 100, 1000. Selanjutnya pengujian fungsi aktivasi ada 3 yaitu *sigmoid*, *sin*, *relu*, untuk *random type* normal, dan untuk jumlah *hidden neuron* adalah 100, 150, 200. Dari Pengujian tersebut akan dicari nilai akurasi yang paling baik.

Dari hasil pengujian yang dilakukan akan ditinjau dari *confusion matrix* dan nilai matrik evaluasi untuk mendapatkan hasil klasifikasi menggunakan metode ELM terbaik.

Berikut adalah hasil perbandingan matrik evaluasi pada data *imbalance* dan *balance* pada setiap fungsi aktivasi dan split data 70%:30% dan 80%:20%.

Pada tabel 2 berikut akan melihat nilai dari matrik evaluasi *sensitivity*, *specificity*, *G-mean* pada data *imbalance* (sebelum smote).

TABEL 2
Hasil evaluasi data *imbalance* (sebelum smote)

Train/Test	Fungsi Aktivasi	Sensitivity	Spesificity	G-mean
70/30	Sigmoid	0.91	0.96	0.94
	Sin	0.91	0.97	0.94
	ReLU	0.91	0.94	0.94
80/20	Sigmoid	0.92	0.97	0.94
	Sin	0.91	0.97	0.94
	ReLU	0.92	0.96	0.94

Sedangkan, pada tabel 3 berikut akan melihat nilai dari matrik evaluasi *sensitivity*, *specificity*, *G-mean* pada data *balance* (setelah smote).

TABEL 3
Hasil evaluasi data *balance* (setelah smote)

Train/Test	Fungsi Aktivasi	Sensitivity	Spesificity	G-mean
70/30	Sigmoid	0.93	0.95	0.94
	Sin	0.92	0.93	0.93
	ReLU	0.90	0.96	0.93
80/20	Sigmoid	0.95	0.98	0.96
	Sin	0.95	0.97	0.96
	ReLU	0.94	0.95	0.94

Dari sekian banyak hasil pengujian, diambil nilai evaluasi paling baik pada setiap fungsi aktivasi yang digunakan. Berikut adalah hasil terbaik dari setiap fungsi aktivasi pada masing-masing percobaan menggunakan data *imbalance* dan *balance*.

1. Klasifikasi menggunakan data *imbalance*

Proses pengujian klasifikasi menggunakan metode ELM untuk dataset *imbalance*. Data *imbalance* diambil dari kelima stasiun dari 3 kelas baik, sedang, tidak sehat. Diambil masing-masing 500 data, dan kelas sangat tidak sehat sebanyak 30 data, kemudian digabungkan menjadi total data untuk kelima stasiun 1.530 data. Berikut adalah hasil dari percobaan pada dataset ISPU *imbalance*.

Dari proses klasifikasi data *imbalance* akan dikumpulkan nilai akurasi paling baik di setiap fungsi aktivasi percobaan. Pada tabel 4 berikut adalah hasil dari pengujian terbaik data *training* berdasarkan setiap fungsi aktivasi.

TABEL 4
Pengujian terbaik data *training* pada data *imbalance*

Activation Function	Split Data	Regularization Parameter	Hidden Neuron	Random Type	Data Training			
					Sensitivity	*Spesificity*	*G-Mean*	Accuracy
Sigmoid	70/30	1000	500	Normal	0.98	0.97	0.97	0.97
Sin	70/30	10	500	Normal	0.99	0.99	0.99	0.99
ReLU	70/30	10	500	Normal	0.99	0.99	0.99	0.99
Best Result					0.99	0.99	0.99	0.99

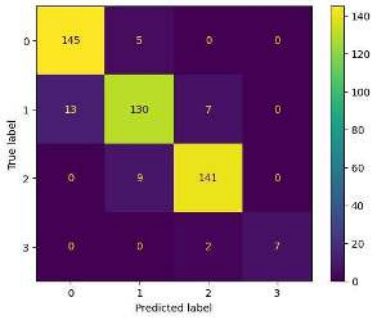
Pada tabel 5 berikut adalah hasil dari pengujian terbaik data *testing* dari setiap lokasi berdasarkan setiap fungsi aktivasi.

TABEL 5
Pengujian terbaik data *testing* pada data *imbalance*

Activation Function	Split Data	Regularization Parameter	Hidden Neuron	Random Type	Data Testing			
					Sensitivity	Spesificity	G-Mean	Accuracy
Sigmoid	70/30	1000	500	Normal	0.91	0.96	0.94	0.94
Sin	70/30	10	500	Normal	0.91	0.97	0.94	0.94
ReLU	70/30	10	500	Normal	0.91	0.97	0.94	0.94
Best Result					0.91	0.97	0.94	0.94

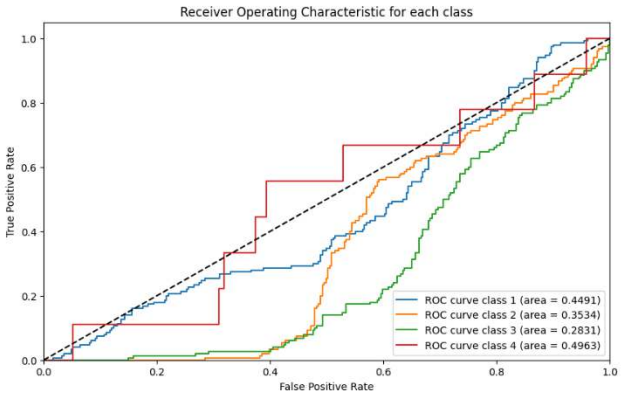
Dari hasil pengujian yang telah dikumpulkan, hasil akurasi paling bagus dengan parameter pengujian yaitu fungsi aktivasi menggunakan ReLU, nilai C 10, split data 70%:30% *random type* normal, dan jumlah hidden neuron 500.

Gambar 2 berikut adalah *confusion matrix* dari hasil pengujian terbaik dataset *imbalance*.



GAMBAR 2
Confusion Matrix pengujian terbaik pada data *imbalance*

Pada gambar 3 berikut ini merupakan gambar hasil dari perhitungan ROC pada proses klasifikasi dengan hasil terbaik menggunakan data *imbalance*.



GAMBAR 3
Hasil ROC pengujian terbaik pada data *imbalance*

2. Klasifikasi menggunakan data *balance*

Proses pengujian klasifikasi menggunakan metode ELM untuk dataset *balance*. Data *balance* diambil dari kelima stasiun dari 4 label baik, sedang, tidak sehat, dan sangat tidak sehat. Diambil masing-masing 500 data kemudian digabungkan menjadi total data untuk kelima stasiun 2.000 data. Berikut adalah hasil dari percobaan pada dataset ISPU *balance*:

Dari proses klasifikasi menggunakan data *balance* akan dikumpulkan nilai akurasi paling baik di setiap fungsi aktivasi percobaan. Pada tabel 6 berikut adalah hasil dari pengujian terbaik data *training* berdasarkan setiap fungsi aktivasi.

TABEL 6
Pengujian terbaik data *training* pada data *balance*

Activation Function	Split Data	Regularization Parameter	Hidden Neuron	Random Type	Data Training			
					Sensitivity	Spesificity	G-Mean	Accuracy
Sigmoid	80/20	100	500	Normal	0.95	0.96	0.95	0.95
Sin	80/20	10	300	Normal	0.95	0.98	0.96	0.96
ReLU	80/20	100	500	Normal	0.98	0.99	0.98	0.98
Best Result					0.95	0.96	0.95	0.95

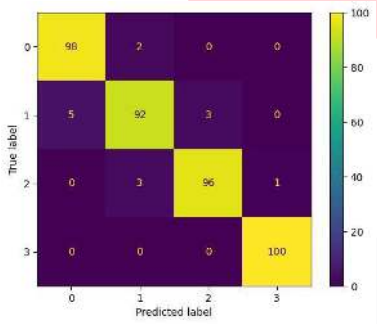
Pada tabel 7 berikut adalah hasil pengujian terbaik data *testing* berdasarkan setiap fungsi aktivasi.

TABEL 7
Pengujian terbaik data *testing* pada data *balance*

Activation Function	Split Data	Regularization Parameter	Hidden Neuron	Random Type	Data Testing			
					Sensitivity	Spesificity	G-Mean	Accuracy
Sigmoid	80/20	100	500	Normal	0.95	0.98	0.96	0.96
Sin	80/20	10	500	Normal	0.95	0.97	0.96	0.96
ReLU	80/20	100	500	Normal	0.94	0.95	0.94	0.94
Best Result					0.95	0.98	0.96	0.96

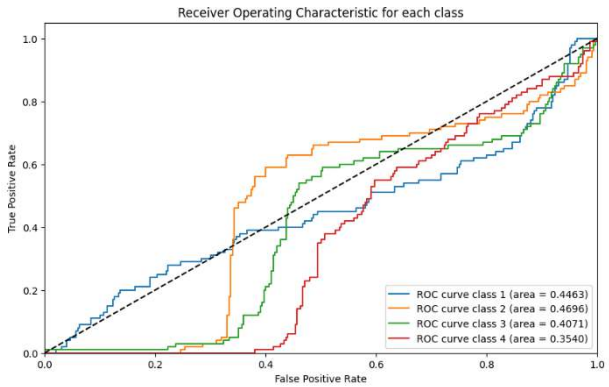
Dari hasil pengujian yang telah dikumpulkan, hasil akurasi paling bagus dengan parameter pengujian yaitu fungsi aktivasi menggunakan *Sigmoid* nilai C 100 , split data 80%:20%, random type normal, dan jumlah hidden neuron 500.

Gambar 4 berikut adalah *confusion matrix* dari hasil pengujian terbaik dataset *balance*.



GAMBAR 4
Confusion Matrix pengujian terbaik pada data *balance*

Pada gambar 5 berikut ini merupakan gambar hasil dari perhitungan ROC pada proses klasifikasi dengan hasil terbaik menggunakan data *balance*.



Gambar 5
Hasil ROC pengujian terbaik pada data *balance*

V. KESIMPULAN

Berdasarkan pengujian dan analisis yang dilakukan untuk klasifikasi dataset ISPU DKI Jakarta dengan menggunakan metode *Extreme Learning Machine* didapatkan hasil yang paling baik untuk klasifikasi pada data *balance*. Hasil tersebut didapatkan berdasarkan analisis *confusion matrix* dan nilai matriks evaluasi, yaitu *Sensitivity* 0.95, *Specificity* 0.96, *G-mean* 0.95, dan *Accuracy* 0.95 untuk data *training*. Sedangkan untuk data *testing* menghasilkan nilai matriks evaluasi, yaitu *Sensitivity* 0.95, *Specificity* 0.98, *G-mean* 0.96, dan *Accuracy* 0.96. Parameter yang digunakan sehingga mendapatkan hasil tersebut yaitu split data 80%:20%, fungsi aktivasi Sigmoid, nilai C pada regularisasi parameter 100, *random type* normal, dan jumlah hidden neuron 500.

REFERENSI

[1] IQAir, "Kualitas Udara di Jakarta," *IQAir*, Oct. 2022. https://www.iqair.com/id/indonesia/jakarta (accessed Jul. 31, 2023).

[2] Gramedia Blog, "Dampak Negatif dari Pencemaran Udara & Solusinya," *Gramedia Blog*, Oct. 24, 2022. https://www.gramedia.com/literasi/dak-negatif-dari-pencemaran-udara (accessed Jul. 31, 2023).

[3] A. Nur Alfiyatin *et al.*, "Penerapan Extreme Learning Machine (ELM) Untuk Peramalan Laju Inflasi Di Indonesia Implementation Extreme Learning Machine For Inflation Forecasting In Indonesia," vol. 6, no. 2, pp. 179–186, 2018, doi: 10.25126/jtiik.20186900.

[4] Daniel A. Vallero, *Fundamentals of Air Pollution*, Fifth Edition. New York: Academic Press, 2014.

[5] A. Budiyono, "Pencemaran Udara : Dampak Pencemaran Udara Pada Lingkungan."

[6] J. Han, M. Kamber, and J. Pei, "Data Mining. Concepts and Techniques, 3rd Edition (The Morgan Kaufmann Series in Data Management Systems)," 2011.

[7] Arwan *et al.*, "Synthetic Minority Over-sampling Technique (SMOTE) Algorithm For Handling Imbalanced Data," *Binus University*, Jakarta, Jun. 08, 2018. Accessed: Aug. 15, 2023. [Online]. Available: https://mti.binus.ac.id/2018/06/08/synthetic-minority-over-sampling-technique-smote-algorithm-for-handling-imbalanced-data/

[8] Andi, "Memahami dan Menerapkan Matriks Evaluasi ROC-AUC dalam Machine Learning," *medium.com*, Aug. 13, 2023. https://medium.com/@andimrinaldisaputraa/memahami-dan-menerapkan-matriks-evaluasi-roc-auc-dalam-machine-learning- (accessed Aug. 15, 2023).

[9] G. Bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006, doi: 10.1016/j.neucom.2005.12.126.

[10] I. P. Siwi, I. Cholisoddin, and T. M. Furqon, "Peramalan Produksi Gula Pasir Menggunakan Extreme Learning Machine (ELM) pada PG Candi Baru Sidoarjo," Universitas Brawijaya, Malang, 2016.