

# PEMODELAN KEDALAMAN LAUT PADA PERAIRAN SELAT SUNDA DAN SEKITARNYA MENGGUNAKAN *NEURAL NETWORK*

Gabriel Sahata Sianturi\* dan Rifky Fauzi

Program Studi Matematika, Institut Teknologi Sumatera, Bandar Lampung

\*Email: gabriel.120160026@student.itera.ac.id

## ABSTRAK

Indonesia telah ditetapkan sebagai negara kepulauan oleh Deklarasi Juanda pada tahun 1957 dan diperkuat oleh United Nations Convention on the Law of the Sea 1982 (UNCLOS 1982). Berbagai upaya telah dilakukan untuk memetakan wilayah maritim Indonesia, salah satunya adalah peta kontur batimetri. Peta analog memiliki keterbatasan karena bentuk fisiknya serta informasi yang belum lengkap. Penelitian ini bertujuan untuk memodelkan peta kedalaman dengan menggunakan *neural network*. Data diperoleh dari digitasi manual dengan mengambil data kedalaman pada peta kontur menggunakan perangkat lunak ArcGIS. Proses digitasi menghasilkan *dataset* sebanyak 41.978 titik yang terdiri dari informasi koordinat dan kedalaman. *Dataset* yang telah didigitalkan ini digunakan sebagai input untuk arsitektur model jaringan saraf untuk memprediksi kedalaman yang sesuai dengan representasi analog. Analisis *error* pada arsitektur model *neural network* menunjukkan nilai kesalahan yang rendah dengan RMSE sebesar 0,00277, menandakan efektivitas model dalam merepresentasikan kedalaman secara akurat sesuai dengan peta analog. Studi ini berkontribusi pada penerapan model jaringan saraf dengan data analog yang telah didigitalkan untuk prediksi kedalaman yang lebih baik.

**Kata kunci:** digitasi, *feedforward neural network*, peta analog.

## 1 PENDAHULUAN

Indonesia ditetapkan sebagai negara kepulauan oleh Deklarasi Juanda pada tahun 1957 dan didukung dengan United Nations Convention on the Law of the Sea 1982 (UNCLOS 1982). Pada Deklarasi Juanda 1957, Indonesia menyatakan kepada dunia bahwa perairan Indonesia (perairan yang mengelilingi, di antara dan di dalam kepulauan Indonesia) telah menjadi satu kesatuan wilayah Negara Kesatuan Republik Indonesia. Berdasarkan UNCLOS 1982 (Sharma, 1995), luas wilayah perairan Indonesia yaitu 5,9 juta km<sup>2</sup> yang meliputi 3,2 juta km<sup>2</sup> perairan teritorial dan 2,7 juta km<sup>2</sup> perairan Zona Ekonomi Eksklusif (ZEE), dan luas wilayah perairan tersebut tidak termasuk dalam landas kontinen (*continental shelf*). Dengan ini, Indonesia disebut sebagai negara kepulauan terbesar di dunia.

Selat Sunda merupakan salah satu wilayah perairan yang ada di Indonesia. Selat Sunda menghubungkan 2 pulau yang ada di Indonesia yaitu Pulau Jawa dan Pulau Sumatera. Selat Sunda memiliki kedalaman minimal yaitu 24 m dan memiliki kedalaman hingga 1200 m (Amri, 2017). Data tentang kedalaman laut memiliki banyak kegunaan salah satunya dapat memprediksi ketinggian gelombang dan input untuk simulasi perambatan gelombang. Dalam (Masrukhin, dkk., 2014) menjelaskan bahwa kegunaan dari data kedalaman yang bisa digunakan untuk menentukan jalur peletakkan pipa bawah laut. Berdasarkan (Lubis, dkk., 2020) menggunakan data kedalaman laut di sekitar Pelabuhan Batu Ampar dengan data satelit dan memanfaatkan data pasang surut sebagai koreksi. Sehingga data tersebut sangat dibutuhkan oleh para peneliti untuk mengembangkan penelitian-penelitian tentang wilayah perairan Indonesia.

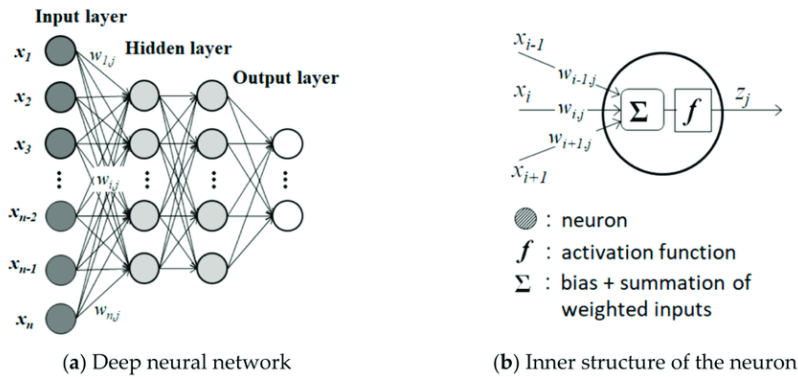
Data kedalaman air laut dapat ditemukan dalam peta kedalaman laut. Peta kedalaman laut memiliki beberapa keterbatasan salah satunya berbentuk fisik atau biasa disebut peta analog. Jika membandingkan data dari GEBCO yang bisa diakses secara bebas dan diketahui kedalaman disetiap titik, sedangkan peta analog hanya diketahui beberapa titik dan disepanjang garis kontur saja. Dari peta analog tersebut, bisa didapatkan data terkait kedalamannya dengan melakukan digitasi pada peta tersebut. Pada (Awaludin, dkk., 2017) data kedalaman yang digunakan diinterpolasi menggunakan *ArcMap 10.2* dengan metode interpolasi yaitu *Topo To Raster* sehingga diperoleh kontur kedalaman tersebut yang dimana ArcMap merupakan fitur dari *software* ArcGIS. Untuk merepresentasikan kontur peta tidak hanya memerlukan data kedalaman laut, dibutuhkan juga titik koordinat dari data kedalaman tersebut agar bisa diolah sehingga bisa merepresentasikan peta kontur. Titik koordinat tersebut dapat diperoleh pada *software* ArcGIS yaitu dengan mendigitasi peta kedalaman laut tersebut.

Data hasil digitasi tersebut berisi data yang cukup besar, maka diperlukannya sebuah solusi untuk mengatasi data yang besar tersebut agar bisa memprediksi nilai kedalaman yang sesuai dengan peta analog tersebut. Salah satunya yaitu dengan menggunakan model jaringan saraf tiruan (*Neural Network*). Model jaringan saraf tiruan dapat membuat model yang efisien dan fleksibel walaupun memiliki data yang cukup besar. Dari latar belakang tersebut, maka artikel ini bertujuan untuk dapat mendigitasi data peta analog menjadi peta digital menggunakan *software* ArcGIS dan dapat menentukan konfigurasi arsitektur model *Neural Network* yang terbaik dengan nilai *error* yang kecil dimana menggambarkan model tersebut dapat memprediksi kedalaman sesuai dengan peta analog.

## 2 METODE

Metode penelitian yang dilakukan menggunakan jaringan syaraf tiruan. Jaringan syaraf tiruan adalah salah satu sistem pemrosesan informasi yang menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot dan proses uji. *Neural network* atau jaringan syaraf tiruan adalah salah satu cabang dari *Artificial Intelegant* (AI) yang memproses informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot (Kiki dan Kusumadewi, 2004). Jaringan syaraf tiruan dapat mengenali kegiatan berbasis pada data masa lalu. Data masa lalu tersebut akan dipelajari oleh jaringan syaraf tiruan yang mempunyai kemampuan untuk memberikan keputusan terhadap data yang belum dipelajari (Hermawan dan Sigit Suyantoro, 2006 dalam Kartini, dkk., 2017).

Model *Neural Network* adalah model untuk permasalahan linier dan nonlinier yang terdiri dari input data, *hidden layer* (terdapat beberapa neuron), dan *output layer*. Pada neuron, terdapat fungsi aktivasi untuk memproses dari input data. Proses model *neural network* dari tahapan input data hingga mendapatkan *output* disajikan pada Gambar 1 (Ghozi, dkk., 2022).



**Gambar 1.** Arsitektur model *neural network*

Karena tidak semua masalah dapat diselesaikan dengan arsitektur yang sama, arsitektur jaringan syaraf tiruan menentukan seberapa sukses target yang akan dicapai. Namun, beberapa parameter dasar tetap digunakan apapun jenis arsitekturnya. *Epoch* dan fungsi aktivasi adalah beberapa parameter tersebut. Fungsi aktivasi adalah fungsi yang digunakan pada jaringan saraf untuk mengaktifkan atau tidak mengaktifkan neuron. Karakteristik yang harus dimiliki oleh fungsi aktivasi jaringan perambatan balik antara lain harus kontinu, terdiferensialkan, dan tidak menurun secara monotonis (*monotonically non-decreasing*) (Julpan, dkk., 2015). Fungsi aktivasi yang terdapat pada model *neural network* ada beberapa fungsi antara lain

a. Fungsi Sigmoid

Fungsi Sigmoid digunakan untuk memperoleh *output* yang bersifat *nonlinier*. Sigmoid akan menerima angka tunggal dan mengubah nilai  $x$  menjadi sebuah nilai yang memiliki *range* mulai dari 0 sampai 1. Sigmoid memiliki bentuk formula sebagai berikut.

$$y(u) = \frac{1}{1 + e^{(-u)}} \quad (1)$$

b. Fungsi ReLU

Fungsi Rectified Linier Unit (ReLU) adalah fungsi yang simpel dikarenakan hanya membuat pembatas pada bilangan nol, artinya apabila  $x < 0$  maka  $x = 0$  dan apabila  $x > 0$  maka  $x = x$ . Fungsi ReLU dinyatakan dengan persamaan berikut.

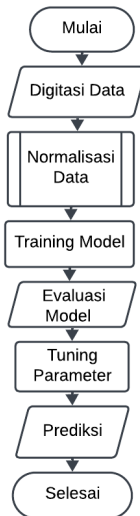
$$y(u) = \max(0, u) \quad (2)$$

c. Fungsi Tanh

Fungsi Tanh akan mengubah nilai input  $x$  nya menjadi sebuah nilai yang memiliki *range* dari -1 sampai 1. Dalam praktiknya Tanh lebih menjadi pilihan utama jika dibandingkan dengan sigmoid. Perlu diketahui, tanh merupakan sigmoid yang dikembangkan. Fungsi Tanh memiliki persamaan sebagai berikut.

$$\tanh(u) = \frac{2}{1 + e^{(-2u)}} - 1 \quad (3)$$

Pada bagian ini, akan diuraikan dalam *flowchart* sebagai berikut.

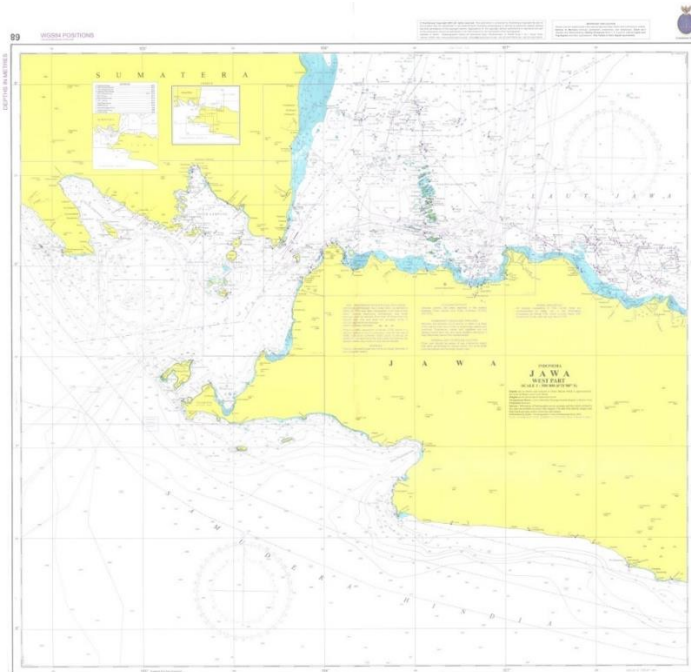


**Gambar 2.** Kerangka kerja penelitian

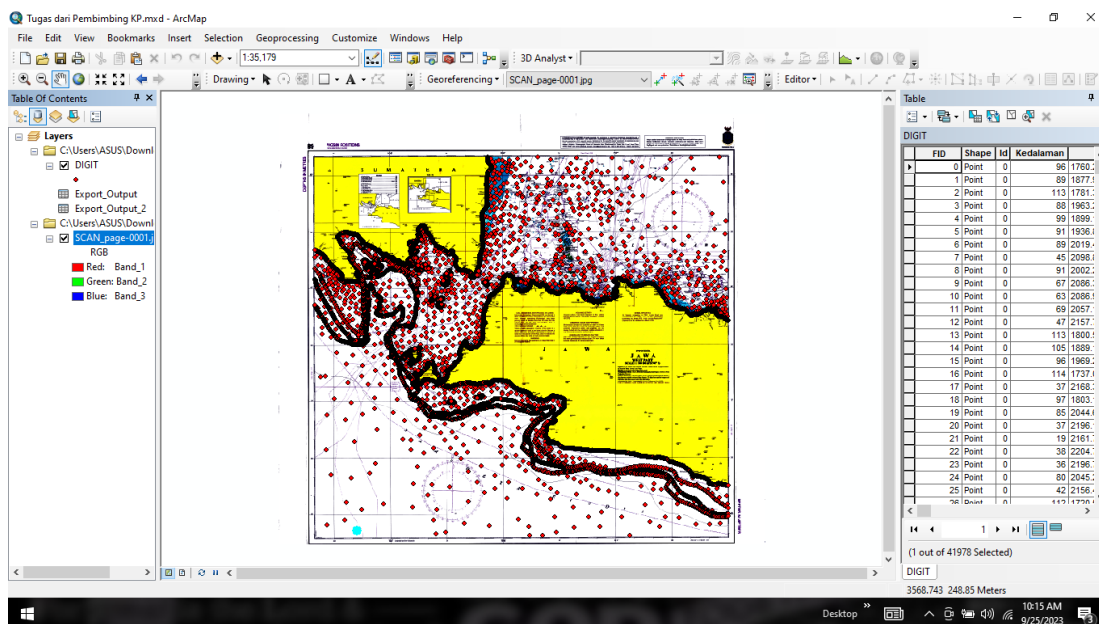
Proses studi ini dilakukan menggunakan *software* ArcGIS dan Bahasa pemrograman *Python* melalui *Googlecolab* mulai dari digitasi data hingga proses prediksi. Modul utama yang digunakan adalah *Keras* dari *Tensorflow*. Sebagai modul pendukung, digunakan modul *Numpy*, *Pandas* dan *Matplotlib*.

## 2.1 Digitasi Data

Data yang digunakan adalah peta kedalaman laut yang berasal dari Pusat Hidro-Oseanografi Angkatan Laut pada tahun 2021 (lihat Gambar 3). Digitasi dilakukan menggunakan *software* ArcGIS. Berikut langkah-langkah digitasi pada *software* ArcGIS. Pertama, persiapkan peta yang akan di-digitasi kemudian *import* peta tersebut kedalam *software* ArcGIS. Kedua, aktifkan Editor di ArcMap atau Editor Tab di ArcGIS. Ketiga, pilih *layer* tempat yang akan didigitasi. Ini mungkin bisa *layer* kosong atau *layer* yang sudah ada. Keempat, pilih alat digitasi yang sesuai untuk jenis objek yang akan di-digitasi (*point*, *line*, atau *polygon*). Kelima, mulailah menempatkan titik atau membuat garis atau poligon dengan mengklik pada lokasi yang sesuai pada peta. Bisa juga menggunakan alat edit untuk memodifikasi digitasi jika diperlukan. Keenam, setelah menyelesaikan digitasi suatu objek, atur atributnya sesuai dengan informasi yang relevan, seperti nama, kategori, atau atribut lainnya. Ketujuh, simpan perubahan yang sudah dibuat menggunakan alat *editor*. Hal tersebut akan menyimpan data digitasi ke dalam *layer*. Kedelapan, teruskan proses digitasi untuk objek berikutnya hingga selesai. Kesembilan, setelah selesai digitasi, hentikan *mode editor* untuk mengkonfirmasi dan menyimpan perubahan secara permanen. Kesepuluh, simpan proyek ArcMap atau ArcGIS untuk menyimpan seluruh konfigurasi dan perubahan yang telah dibuat. Terakhir, dapat meng-*export* data dalam format yang sesuai, seperti *shapefile* atau *geodatabase* pada peta yang sudah di-digitasi. Hasil digitasi dari peta yang digunakan dalam artikel ini disajikan pada Gambar 4.



**Gambar 3.** Peta Kedalaman Selat Sunda dan sekitarnya (Pusat Hidro-Oseanografi Angkatan Laut, 2021)



**Gambar 4.** Hasil digitasi pada *software* ArcGIS

Gambar 3 menjelaskan bahwa hasil peta setelah di-digitasi. Dengan titik-titik tersebut menandakan kedalaman dan titik koordinatnya. Lalu hasil digitasi disajikan pada Tabel 1. Dari Tabel 1, didapatkan data sebanyak 41.978 data yang terdiri dari kedalaman dan titik koordinat. Data tersebut yang akan menjadi input pada model *Feedforward Neural Network*.

**Tabel 1.** Data hasil digitasi

No	Kedalaman	X	Y
----	-----------	---	---

1	96	1760,200662	-2610,16513
2	89	1877,940481	-2557,909817
3	113	1781,367371	-2514,914939
4	88	1963,268776	-2624,717242
5	99	1899,10719	-2469,935683
...	...	...	...
41974	200	1616,344072	-2073,272381
41975	200	1616,740948	-2080,614583
41976	200	1616,013343	-2089,610434
41977	200	1613,301358	-2098,804724
41978	200	1611,184687	-2106,477656

### 1. Normalisasi Data

Normalisasi data dilakukan pada data dengan mentransformasikan data menjadi nilai dengan rentang interval [0,1]. Teknik normalisasi yang digunakan yaitu *min-max scaling* sebagai berikut.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4)$$

Dengan  $x'$  menyatakan data hasil normalisasi,  $x$  menyatakan data yang akan dinormalisasi,  $\max(x)$  menyatakan nilai terbesar pada data, dan  $\min(x)$  menyatakan nilai terkecil pada data

### 2. Melatih Model

Model dibangun dengan menggunakan data dengan membagi data menjadi dua yaitu data latih dan data tes. Data latih membantu algoritma belajar menggunakan model, sedangkan data tes menentukan seberapa baik algoritma bekerja pada model, sehingga algoritma tersebut dapat menghitung setelah memasukkan data baru. Perbandingan data latih dan data tes pada penelitian ini adalah 80 % untuk data latih dan 20% untuk data tes.

### 3. Evaluasi Data Latih

Model tersebut akan dilatih dengan memanfaatkan fungsi loss berupa *root mean square error* (RMSE).

$$Loss = \sqrt{\frac{\sum_{i=1}^N (t_i - y_i)^2}{N}} \quad (5)$$

Misal  $N$  menyatakan banyak data,  $t$  menyatakan data target,  $y$  menyatakan *output* model.

### 4. Tuning Parameter

*Tuning* parameter dilakukan untuk menyesuaikan hasil pelatihan dan prediksi yang didapatkan agar lebih optimal. Parameter yang dituning yaitu banyaknya *epoch*, *hidden nodes*, *test size*, dan *neuron*.

### 5. Prediksi

Hasil prediksi yang optimal pada model dapat dilihat dari nilai RMSE *train* dan RMSE *test* yang terkecil.

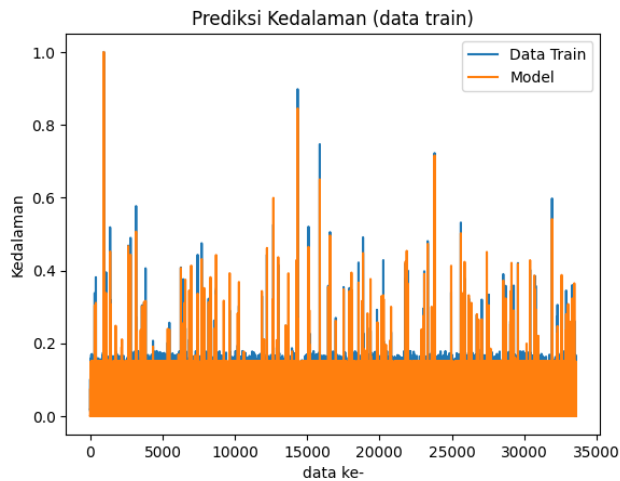
### 3 HASIL DAN PEMBAHASAN

Hal pertama yang dilakukan untuk prediksi dengan *feedforward neural network* adalah dengan membuat inisialisasi jaringan. Sebelum melakukan latihan, nilai parameter yang diinginkan ditentukan terlebih dahulu agar memperoleh hasil yang optimal.

**Tabel 2.** Arsitektur *feedforward neural network*

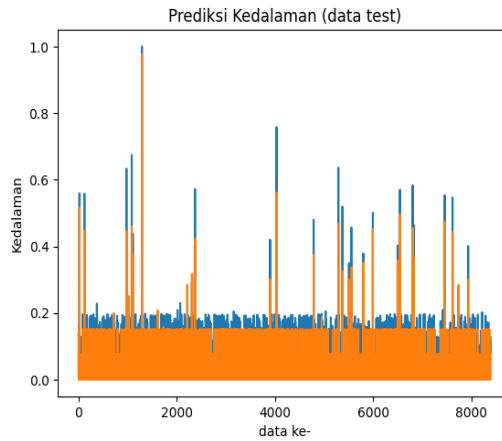
Karakteristik	Spesifikasi
Arsitektur	3 Hidden Layer Dense
Neuron Input	2
Neuron Hidden Dense 1	128
Neuron Hidden Dense 2	64
Neuron Hidden Dense 3	20
Fungsi Aktivasi	ReLU, ReLU, Sigmoid
Epoch	30

Dengan konfigurasi model seperti Tabel 2, didapatkan bahwa nilai RMSE *train* dan RMSE *test* pada data kedalaman laut cukup kecil dengan nilai 0,00020 dan 0,00019. Akan tetapi, dengan melihat hasil plot grafik dari nilai RMSE *train* dan RMSE *test* masih belum sesuai dengan data yang ada. Berikut untuk hasil plot grafik RMSE *train* dan RMSE *test*.

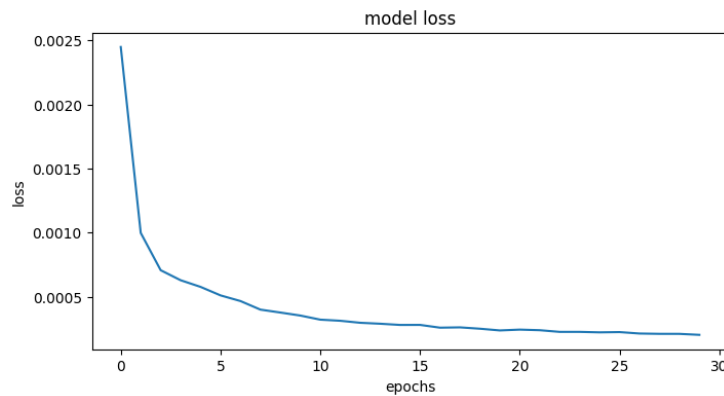


**Gambar 5.** Plot prediksi (data *train*)

Gambar 3 menjelaskan bahwa hasil plot prediksi kedalaman laut menggunakan data *train* sudah menggambarkan data kedalaman yang sesuai dengan peta aslinya. Akan tetapi, hal tersebut nilainya masih dapat berubah-ubah sehingga diperlukannya *tuning*.



**Gambar 6.** Plot prediksi (data tes)



**Gambar 7.** Plot model Loss

Gambar 5 menjelaskan bahwa hasil plot grafik RMSE dan model Loss di atas, diperlukannya *tuning* parameter untuk mendapatkan hasil yang lebih optimum agar model yang dibuat bisa menggambarkan data yang digunakan. Berikut hasil *tuning* dengan 1 *hidden layer*.

Tabel 3 menjelaskan bahwa hasil *tuning* dengan 1 *hidden layer*, didapatkan bahwa nilai RMSE terkecil diperoleh dengan konfigurasi 8 *neuron*, 15 *epoch*, dan 0.00290 untuk nilai RMSE. Kemudian, akan dituning kembali dengan 2 *hidden layer*. Berikut hasil *tuning* menggunakan 2 *Hidden Layer (HL)*.

**Tabel 3.** Hasil *tuning* dengan 1 *hidden layer*

Neuron	Epoch	RMSE	Neuron	Epoch	RMSE
8	10	0,00341	8	35	0,00311
16		0,00362	16		0,00374
32		0,00375	32		0,00384
64		0,00412	64		0,00379
128		0,00405	128		0,00392
256		0,00405	256		0,00395
<b>8</b>	<b>15</b>	<b>0,00290</b>	8	45	0,00358
16		0,00290	16		0,00352



Neuron	Epoch	RMSE	Neuron	Epoch	RMSE
32		0,00350	32		0,00371
64		0,00371	64		0,00392
128		0,00393	128		0,00398
256		0,00403	256		0,00453
8	25	0,00292			
16		0,00356			
32		0,00371			
64		0,00386			
128		0,00406			
256		0,00411			

**Tabel 4.** Hasil *tuning* dengan 2 *hidden layer*

HL 1	HL 2	Epoch	RMSE	HL 1	HL 2	Epoch	RMSE
8	10		0,00422	8	100		0,00417
16			0,00406	16			0,00419
32			0,00406	32			0,00427
64			0,00410	64			0,00451
128			0,00441	128			0,00451
256			0,00414	256		<b>15</b>	0,00469
8	25		0,00408	<b>8</b>	<b>200</b>		<b>0,00394</b>
16			0,00421	16			0,00421
32			0,00404	32			0,00439
64			0,00419	64			0,00458
128			0,00430	128			0,00456
256			0,00441	256			0,00483
8	50		0,00414				
16			0,00407				
32			0,00422				
64			0,00410				
128			0,00409				
256			0,00434				

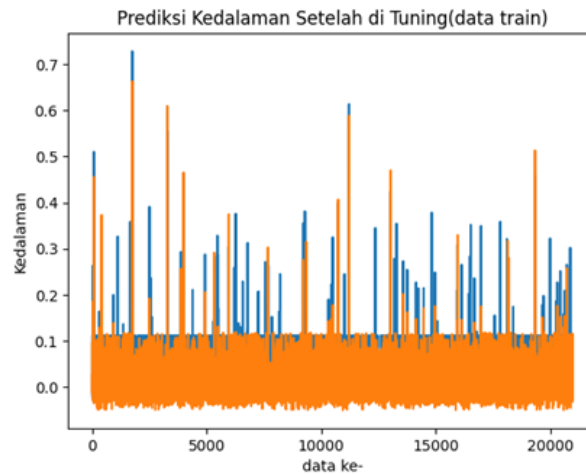
Tabel 4 menjelaskan bahwa konfigurasi model *Feedforward Neural Network* terbaik dengan nilai RMSE terkecil adalah *hidden layer 1* dengan 8 *neuron*, *hidden layer 2* dengan 200 *neuron*, 15 *epoch*, dan 0,00394 untuk nilai RMSE. Dari *tuning* menggunakan 1 *hidden layer* dan 2 *hidden layer* didapatkan bahwa nilai RMSE dengan 1 *hidden layer* mendapatkan nilai RMSE terkecil. Kemudian akan dilakukan *tuning* dengan berbagai *test size* untuk 1 *Hidden Layer*.

**Tabel 5.** Hasil *tuning* dengan berbagai *test size*

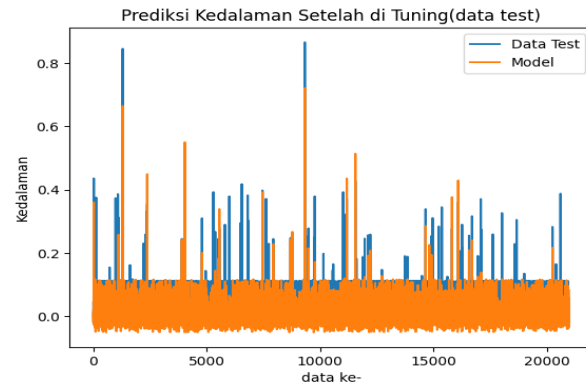
<i>Test Size</i>	RMSE
0,1	0,00393
0,2	0,00289
0,3	0,00326
0,4	0,00284

0,5	0,00277
0,6	0,00279
0,7	0,00292
0,8	0,00291

Tabel 5 menjelaskan bahwa hasil *tuning* dengan berbagai *test size* pada konfigurasi model 1 *hidden layer* mendapatkan nilai RMSE terkecil yaitu 0,00277 dan *test size* sebesar 0,5 yang berarti data *train* dan data *test* memiliki perbandingan yang sama yaitu 50:50. Kemudian akan di-*ploting* grafik hasil *tuning* model dengan 1 *hidden layer*, *test size* 0,5, 8 *neuron*, dan 15 *epoch* sebagai hasil terbaik.



**Gambar 8.** Hasil *tuning* terbaik (data *train*)



**Gambar 9.** Hasil *tuning* terbaik (data *test*)

Gambar 8 dan 9 menggambarkan perbandingan data yang digunakan dan model yang dibuat, bahwa model *feedforward neural network* dapat menggambarkan atau memprediksi kedalaman laut yang sesuai dengan peta yang digunakan sebelumnya.

#### 4 KESIMPULAN

Berdasarkan pengujian arsitektur model *feedforward neural network* dalam memodelkan kedalaman laut Selat Sunda dan sekitarnya, diperoleh hasil performansi model dengan kesimpulan sebagai berikut.

- a. Dapat mendigitasi peta analog menjadi peta digital dan mendapatkan data kedalaman sebanyak 41.978 data dari peta analog menggunakan *software* ArcGIS.
- b. Dapat menentukan konfigurasi arsitektur pada model jaringan saraf tiruan yaitu dengan memiliki input 41.978 data, 1 lapisan tersembunyi (*hidden layer*) dengan 8 *neuron*, *epoch* sebanyak 15, dan *test size* sebesar 0,5, dan 1 lapisan luaran yang menunjukkan nilai prediksi dengan nilai *error* (RMSE) sebesar 0,00277. Hasil *error* tersebut dapat menggambarkan bahwa konfigurasi arsitektur model tersebut dapat memodelkan nilai kedalaman dengan baik.

## UCAPAN TERIMA KASIH

Penulis berterima kasih kepada Dr. Arnida L. Latifah selaku Pembimbing Lapangan Kerja Praktik di Pusat Riset Komputasi, BRIN Cibinong dan penulis juga berterima kasih kepada Jurusan Sains Institut Teknologi Sumatera atas Hibah Penelitian dengan Kontrak No. 8353/IT9.3.1/PT.01.03/2023.

## DAFTAR PUSTAKA

- Amri, K. (2017). Analisis Hubungan Kondisi Oseanografi dengan Fluktuasi Hasil Tangkapan Ikan Pelagis di Selat Sunda. *Jurnal Penelitian Perikanan Indonesia*, 14(1), 55. <https://doi.org/10.15578/jppi.14.1.2008.55-65>.
- Awaludin, A., Zainuri, M., & Sugianto, D. N. (2017). Pemetaan Batimetri dan Sedimen Dasar Segara Anakan Kabupaten Cilacap. *Jurnal Oseanografi*, 6(3), 396–404.
- Ghozi, A. A., Aprianti, A., Dimas, A. D. P., & Fauzi, R. (2022). Analisis Prediksi Data Kasus Covid-19 di Provinsi Lampung Menggunakan Recurrent Neural Network (RNN). *Indonesian Journal of Applied Mathematics*, 2(1), 25. <https://doi.org/10.35472/indojam.v2i1.763>
- Hermawan, A., & Sigit Suyantoro, F. (2006). *Jaringan Saraf Tiruan: Teori dan Aplikasi*.
- Julpan, Nababan, E. B., & Zarlis, M. (2015). Analisis Fungsi Aktivasi Sigmoid Biner dan Sigmoid Bipolar dalam Algoritma Backpropagation pada Prediksi Kemampuan Siswa. *Jurnal Teknovasi*, 02, 103–116.
- Kartini, D., Yani Km, J. A., & selatan, K. (2017). Penerapan Data Mining dengan Algoritma Neural Network (Backpropagation) untuk Prediksi Lama Studi Mahasiswa. *Prosiding Seminar Nasional Sisfotek*, 3584, 235–241.
- Kiki, K., & Kusumadewi, S. (2004). Jaringan Saraf Tiruan dengan Metode Backpropagation untuk Mendeteksi Gangguan Psikologi. *Media Informatika*, 2(2), 1–11. <https://doi.org/10.20885/informatika.vol2.iss2.art1>.
- Lubis, M. Z., Puspita, W. R., Budiana, B., Purba, J. H., & Hakim, R. (2020). Identifikasi Kedalaman Perairan (Batimetri) Terhadap Nilai Kedalaman Data Satelit di Perairan Batu Ampar, Batam. *Journal of Applied Sciences, Electrical Engineering and Computer Technology*, 1(2), 6–12. <https://doi.org/10.30871/aseect.v1i2.2356>.
- Masrukhin, M. A. A., Sugianto, D. N., & Satriadi, A. (2014). Studi Batimetri dan Morfologi Dasar Laut dalam Penentuan Jalur Peletakan Pipa Bawah Laut (Perairan Larangan-Maribaya, Kabupaten Tegal). *Jurnal Oseanografi*, 3(1), 94–104. <http://ejournal-s1.undip.ac.id/index.php/jose>.
- Sharma, O. P. (1995). India and The United Nations Convention on The Law of The Sea. *Ocean Development and International Law*, 26(4), 391–412. <https://doi.org/10.1080/00908329509546068>.