



Prediksi Kecepatan Gelombang S dengan *Machine Learning* pada Sumur “S-1”, Cekungan Sumatera Tengah Indonesia

Sthevanie Dhita Sudrazat¹⁾, Humbang Purba²⁾, Egie Wijaksono²⁾,
Waskito Pranowo¹⁾, dan Muhammad Irsyad Hibatullah²⁾

¹⁾Universitas Pertamina.

Jl. Teuku Nyak Arief, Simprug, Kebayoran Lama, Jakarta

²⁾Pusat Penelitian dan Pengembangan Teknologi Minyak dan Gas Bumi “LEMIGAS”
Jl. Ciledug Raya Kav. 109, Cipulir, Kebayoran Lama, Jakarta Selatan 12230 Indonesia

Artikel Info:

Naskah Diterima:
8 Januari 2020

Diterima setelah
perbaikan:
13 April 2020

Disetujui terbit:
30 April 2020

Kata Kunci:

Kecepatan gelombang S
machine learning
algoritma
parameter

ABSTRAK

Data kecepatan gelombang S (*shear*) sangat diperlukan untuk karakterisasi reservoir dalam menentukan zona reservoir. Namun data kecepatan gelombang S sangat terbatas dan tersedia pada sumur tertentu saja. Penelitian ini dilakukan untuk memprediksi nilai kecepatan gelombang S dengan menggunakan metode *supervised machine learning* pada sumur S-1 lapangan migas di cekungan Sumatra Tengah. Simulasi algoritma *machine learning* dilakukan melalui tahapan sebelum dan setelah *tuning* pada algoritma *library Scikit learn* dan algoritma *artificial neural network* (ANN). Selain itu, parameter dan jumlah data yang digunakan dalam memprediksi nilai kecepatan gelombang akan menentukan nilai *error* dan akurasi. Hasil analisis menunjukkan bahwa algoritma yang digunakan untuk memperoleh akurasi terbaik pertama dalam memprediksi kecepatan gelombang S, yaitu *random forest* dengan nilai parameter *n_estimator* terbaik 10 dan algoritma kedua yang terbaik yaitu *k-nearest neighbor* dengan nilai parameter *n_neighbor* terbaik 5.

© LPMGB - 2020

PENDAHULUAN

Cekungan Sumatera Tengah merupakan cekungan yang telah matang dan termasuk kedalam salah satu cekungan dengan cadangan hidrokarbon terbesar di Indonesia. Analisis lebih lanjut untuk mengetahui properti fisik batuan khususnya pada zona reservoir seperti porositas, litologi dan lain sebagainya perlu dilakukan melalui berbagai pendekatan seperti inversi, multiatribut dan lain sebagainya. Namun terdapat keterbatasan data pada cekungan tersebut

yang diperlukan dalam melakukan karakterisasi reservoir maupun analisis fisika batuan. Salah satu data yang tidak tersedia adalah data kecepatan gelombang S, dimana hanya terdapat satu sumur yang memiliki data tersebut. Oleh karena itu, perlu dilakukan prediksi data kecepatan gelombang S, salah satunya adalah dengan menggunakan *machine learning*.

Penelitian ini bertujuan untuk melakukan estimasi nilai kecepatan gelombang S pada sumur “S-1” dengan menggunakan algoritma *supervised machine learning*. Serta menentukan algoritma *machine learning* terbaik yang sesuai untuk prediksi kecepatan gelombang S pada sumur “S-1”.

Korespondensi:

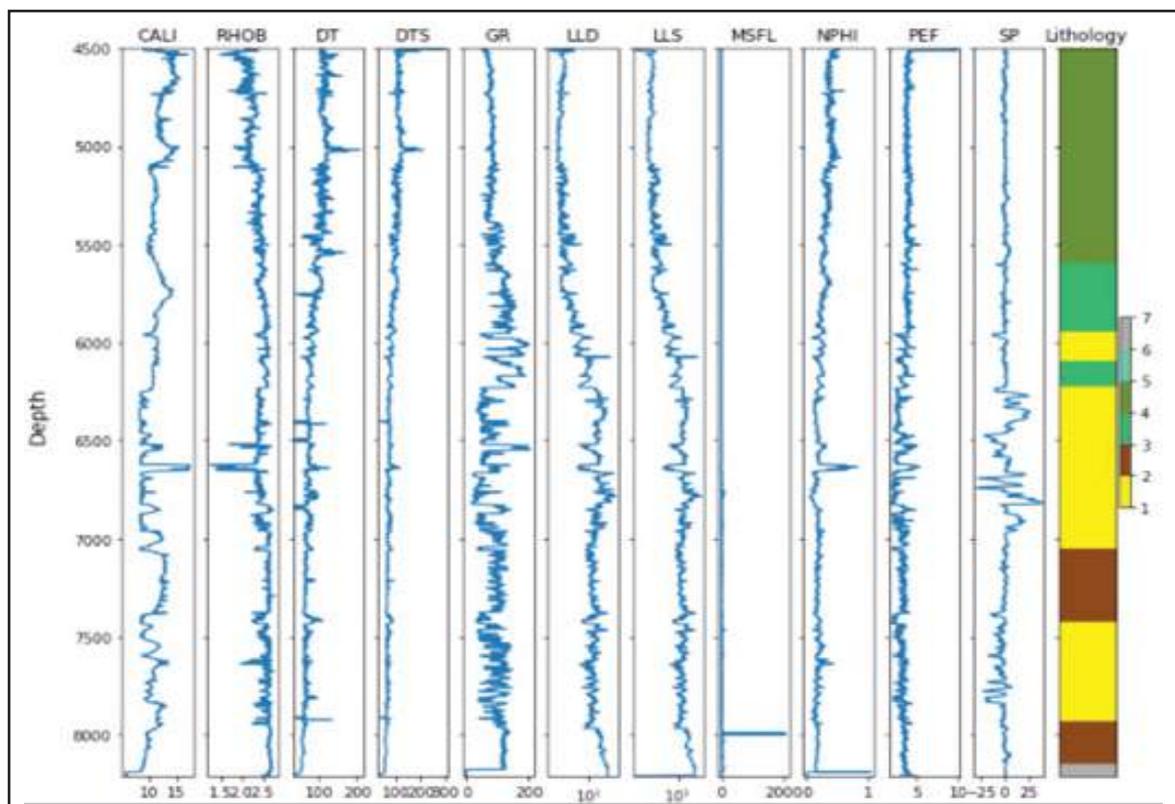
E-mail: sthevy04@gmail.com (Sthevanie Dhita S.)

BAHAN DAN METODE

Penelitian menggunakan data log sumur “S-1” pada Cekungan Sumatera Tengah, dimana merupakan satu-satunya sumur yang memiliki data kecepatan gelombang S. Log sumur yang tersedia antara lain adalah *Depth*, *Caliper*, *Gamma Ray*, *Bulk Density*, *Neutron Porosity*, *Deep Resistivity*, *Shallow Resistivity*, *MSFL*, *Photoelectric Factor*, *Self Potential*, *Sonic compressional (Vp)*, dan *Shear Sonic (Vs)* untuk kedalaman 920 - 8210 ft dengan interval kedalaman 0.5 ft. Kemudian ditambahkan data litologi sebagai data masukan.

Supervised machine learning membutuhkan target data dalam perhitungannya (Somvanshi, dkk., 2016), pada penelitian ini kecepatan gelombang S yang tersedia pada sumur “S-1” menjadi data target. Pengolahan data dilakukan dengan algoritma *K-nearest neighbor (KNN)*, *Support Vector Regressor (SVR)*, *Decision tree*, dan *Random forest* pada *scikit-learn (sklearn)* serta ANN. Pada penelitian ini, penulisan *code script* merujuk pada dokumentasi dari *scikit learn (Scikit-learn developers, 2019)* dan dokumentasi *Keras (fchollet, 2020; Keras.io, 2020)* beserta contohnya.

Terdapat perbedaan pengolahan dengan algoritma *sklearn* dan ANN yang terletak pada *hyperparameter optimization* menggunakan *gridsearch* (Wu, dkk., 2019) yang hanya dilakukan pada metode dengan algoritma pada *sklearn*. Proses lainnya sama, dimana data yang digunakan adalah data sumur “S-1” yang dilakukan perhitungan korelasi terhadap data target dalam hal ini adalah kecepatan gelombang S. Data kemudian diurutkan berdasarkan nilai korelasinya. Data yang digunakan sebagai *input parameter* adalah data dengan nilai korelasi absolut lebih dari 0.5 (Gambar 2) yaitu kecepatan gelombang p, kedalaman, *neutron porosity*, *shallow resistivity*, *bulk density*, *deep resistivity*, *litologi*, dan *photoelectric factor*. *Input data* kemudian dibagi menjadi *training data* dan *test data* dengan komposisi pembagian 30% sebagai *test data* sedangkan 70% sebagai *train data* (Huang, dkk., 2006). Kedua data tersebut dilakukan normalisasi terlebih dahulu sebelum melakukan *training model*. *Training* dilakukan menggunakan *parameter default* dari algoritma. Parameter yang digunakan dalam *training model* digunakan dalam melakukan prediksi pada *test data*. Setelah itu, dilakukan prediksi kecepatan gelombang S untuk



Gambar 1
Visualisasi data log sumur “S-1”.

seluruh sumur. Untuk algoritma ANN hasil prediksi tersebut sudah sebagai hasil akhir. Sedangkan pada algoritma dengan sklearn dilakukan *hyperparameter optimization* untuk mendapatkan parameter lebih baik yang kemudian parameter tersebut digunakan kembali dalam proses *training model* dan *test data* sebelum akhirnya digunakan untuk melakukan prediksi pada keseluruhan data sumur yang tersedia. Prediksi setelah *optimization* tersebut yang menjadi hasil akhir berupa kecepatan gelombang S pada sumur "S-1".

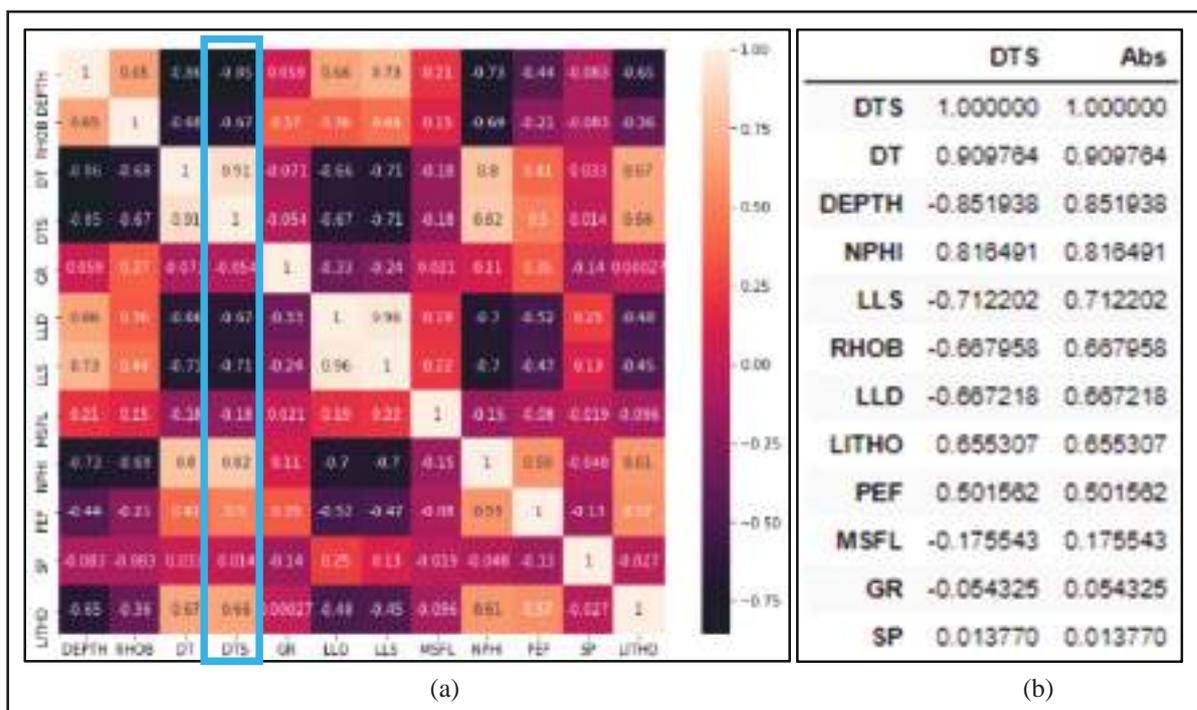
HASIL DAN DISKUSI

Prediksi nilai kecepatan gelombang S dilakukan untuk masing-masing algoritma *machine learning* dengan kedelapan data log sebagai data masukan. Algoritma dalam sklearn dilakukan *hyperparameter optimization (tuning)* sehingga terdapat dua hasil prediksi yaitu sebelum dan sesudah *tuning* yang kemudian dilakukan perbandingan. Sedangkan algoritma ANN tidak dilakukan *tuning* sehingga hanya terdapat satu hasil prediksi.

Prediksi nilai kecepatan gelombang S untuk seluruh sumur "S-1" dengan algoritma ANN menghasilkan nilai yang cukup baik. *Model loss* (Gambar 3) pada *training model* menunjukkan bahwa

algoritma sudah *goodfit* dimana selisih antara *loss* dan *validation loss* mendekati nol. Plot kecepatan gelombang S terhadap kedalaman (Gambar 4) menunjukkan bahwa hampir seluruh nilai dapat terprediksi dengan baik. Walaupun pada *crossplot* antara nilai kecepatan gelombang S *measured* dan hasil prediksi (Gambar 4) menunjukkan nilai kecepatan gelombang S di atas 150 us/ft kurang dapat terprediksi dengan baik.

Hyperparameter optimization (tuning) pada algoritma sklearn dilakukan evaluasi dalam bentuk grafik untuk masing-masing algoritma seperti yang ditunjukkan oleh Gambar 5. Grafik tersebut menunjukkan bahwa algoritma SVR mengalami *underfitting* terlihat dari nilai *error* dari *train* lebih besar dari nilai *error* dari *test*. Kemungkinan hal tersebut diakibatkan oleh kurangnya data masukkan atau rentang nilai parameter C yang belum sesuai. Sedangkan algoritma KNN menunjukkan grafik yang *overfitting* sampai pada nilai *n_neighbor* 8, kemudian *good fit* lalu terjadi *underfitting* pada nilai *n_neighbor* lebih dari 20. Algoritma *decision tree* menunjukkan grafik yang *overfitting* mulai dari nilai *max_depth* lebih dari 5. Pada algoritma *random forest*, grafik menunjukkan bahwa *overfitting* terjadi mulai dari nilai *n_estimator* = 0 dimana selisih antara nilai *train* dan *test error* cukup jauh. Parameter terbaik yang digunakan adalah parameter dengan grafik *goodfit* pada grafik evaluasi *tuning*. Pada algoritma



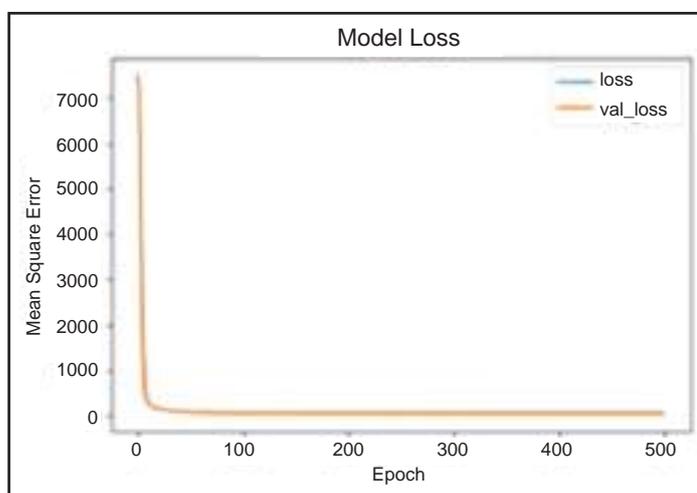
Gambar 2
 (a) Korelasi *heatmap*; (b) tabel hasil korelasi *heatmap*.

KNN, nilai $n_neighbor$ paling baik adalah 10, dimana memiliki grafik yang *goodfit*. Kemudian untuk algoritma SVR digunakan nilai parameter C yang direkomendasikan oleh *grid search* yaitu 29 yang memiliki *error* paling kecil. Pada algoritma *decision tree* digunakan nilai max_depth 5 karena merupakan titik dengan *error* paling kecil sebelum terjadinya *overfitting*. Sedangkan pada algoritma *random forest*, nilai $n_estimator$ yang digunakan adalah 5 yang merupakan nilai *test error* terendah sebelum mengalami kenaikan nilai *error*.

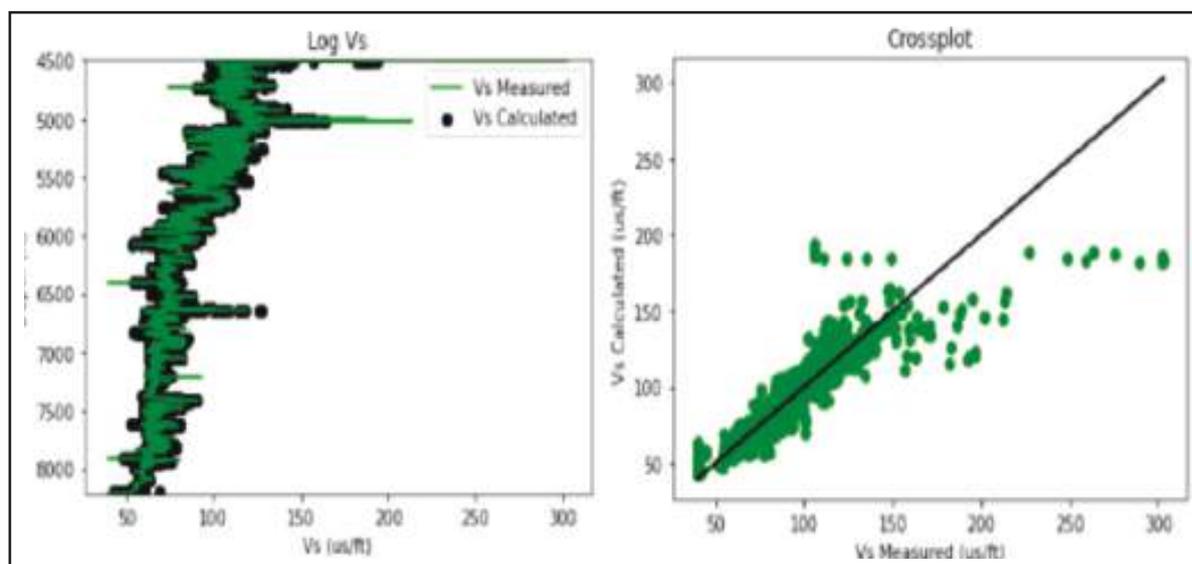
Tabel 1 menunjukkan perbandingan *error* dan akurasi hasil prediksi sebelum dan sesudah *tuning* masing-masing algoritma. Sebagian besar algoritma mengalami penurunan nilai akurasi setelah *tuning*. Hal tersebut terjadi karena sebelum *tuning* parameter yang digunakan merupakan parameter yang *overfitting* dari *training model* sedangkan 70% dari data target merupakan *training data*. Sehingga apabila pada *training model* terjadi *overfitting*, hasil prediksi cenderung sangat mendekati nilai *training data* (Maulana, dkk., 2019) yang juga merupakan data target, namun model menjadi kurang *general* untuk dilakukan pada sumur lain di sekitarnya.

Berdasarkan tabel tersebut, algoritma yang memiliki nilai *error* yang cenderung konsisten bernilai kecil/rendah baik MAE maupun RMS adalah algoritma *random forest*, serta nilai akurasi (R2) juga cukup tinggi.

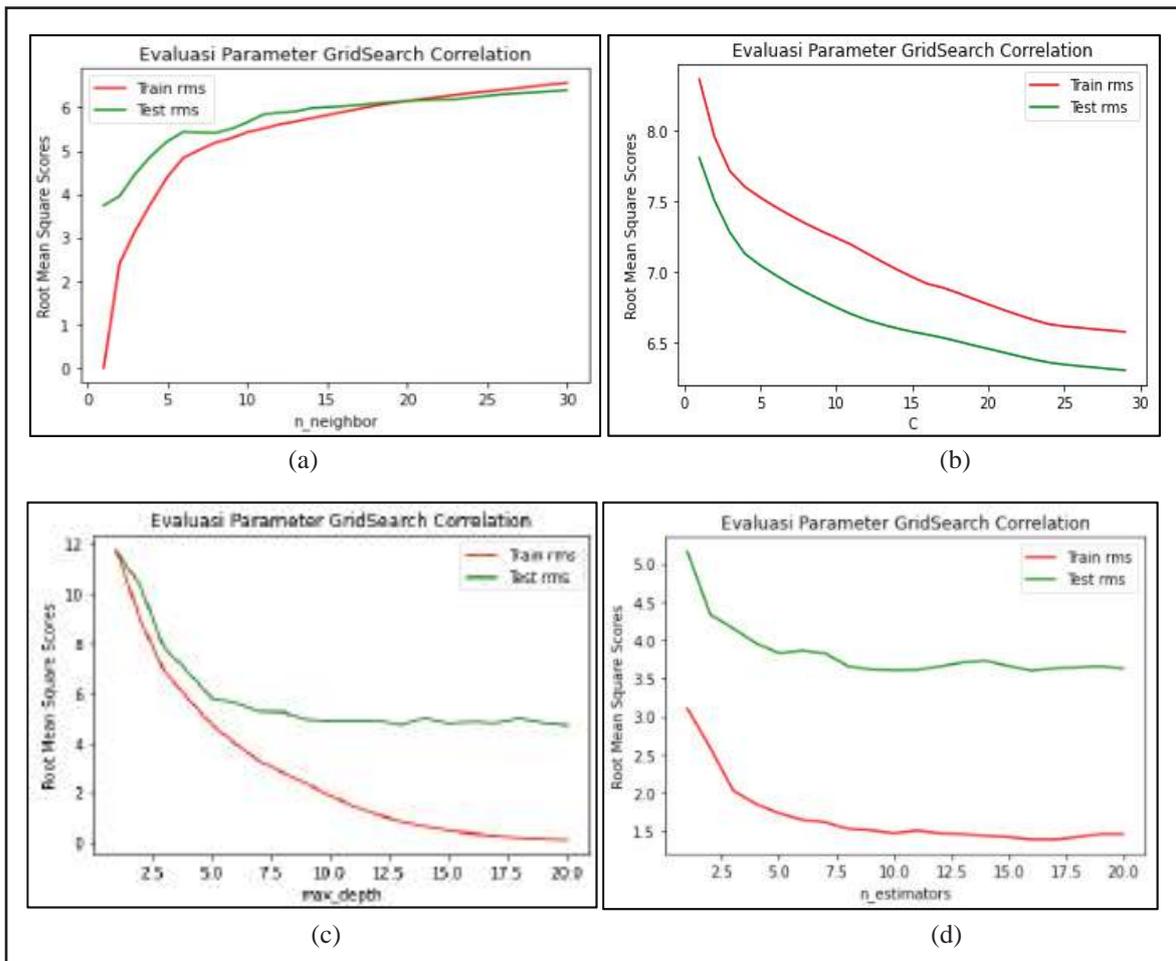
Sedangkan algoritma kedua yang terbaik adalah *k-nearest neighbor*. Meskipun mengalami kenaikan nilai *error* MAE dan RMS, namun kenaikan tersebut tidak terlalu signifikan dibandingkan dengan algoritma lain. Penurunan akurasi antara sebelum dan sesudah *tuning* kedua algoritma juga tidak terlalu signifikan. Sedangkan algoritma SVR walaupun mengalami penurunan nilai *error* dan kenaikan nilai akurasi setelah *tuning*, namun hasil evaluasi parameter menunjukkan nilai yang masih *underfitting* (Gambar 2) dan nilai kecepatan hasil prediksi di beberapa kedalaman tidak mendekati nilai sebenarnya (Gambar 6b), sehingga algoritma ini masih kurang baik digunakan. Algoritma ANN memiliki akurasi yang cukup rendah dibandingkan



Gambar 3
Grafik *model loss* algoritma ANN.



Gambar 4
Hasil prediksi dan *crossplot* algoritma ANN.



Gambar 5
Grafik evaluasi *hyperparameter optimization*: (a) K-Nearest Neighbor; (b) Support Vector Regressor; (c) *Decision Tree*; (d) *Random Forest*.

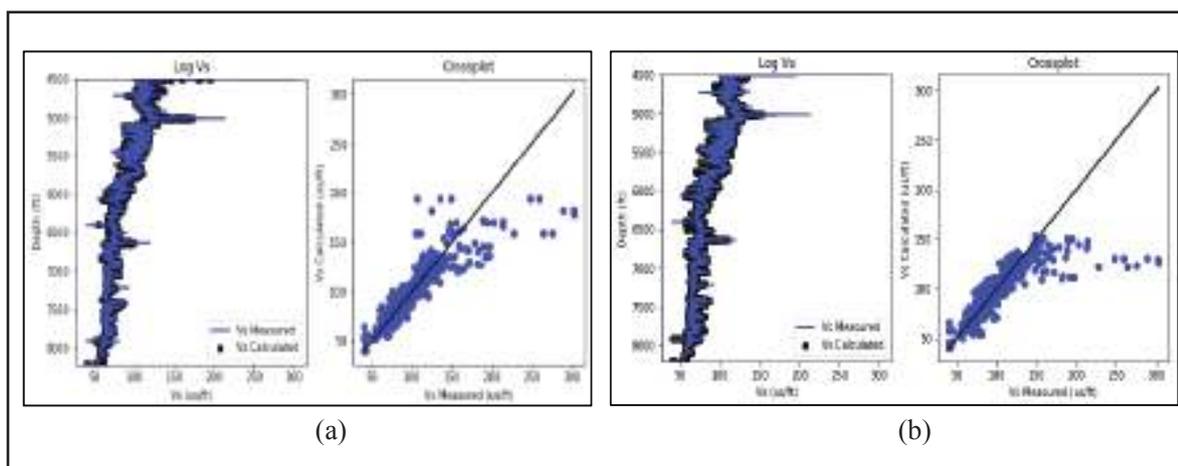
Tabel 1
Perbandingan nilai *error* dan akurasi sebelum dan sesudah *tuning* seluruh algoritma

Algoritma	Sebelum <i>Tuning</i>			Setelah <i>Tuning</i>				
	MAE	RMS	R2	Default param	MAE	RMS	R2	Best param
K-Nearest Neighbor	1.66	5.04	0.948	5	2.03	5.67	0.935	10
SVR	2.44	8.95	0.837	100	1.69	6.87	0.904	29
Random Forest	1.08	5.04	0.948	100	1.24	5.75	0.933	5
Decision Tree	1.08	4.79	0.953	None	2.56	7.38	0.88	5
ANN	2.5	6.35	0.918					

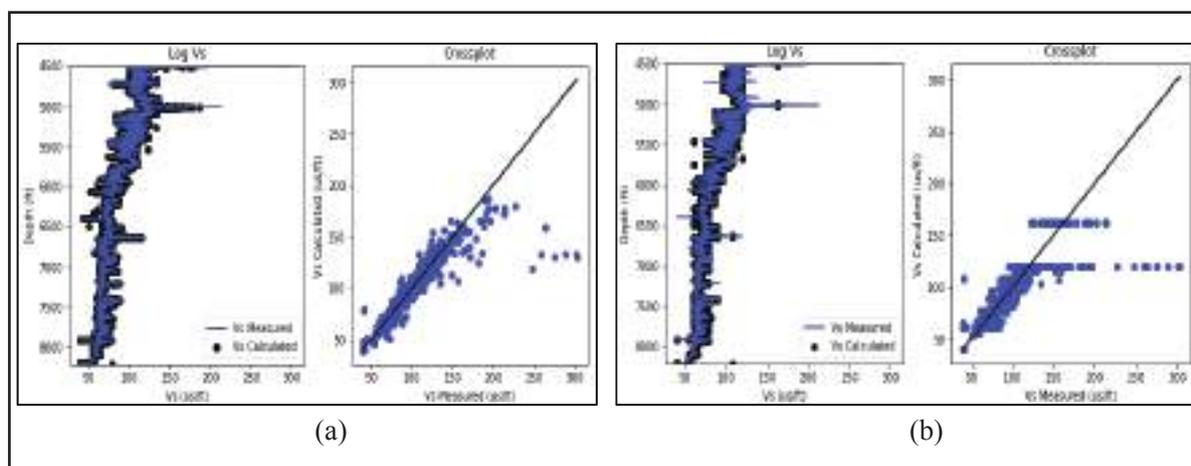
dengan algoritma lain yang menunjukkan bahwa algoritma tersebut belum cukup baik memprediksi nilai kecepatan gelombang. Salah satu faktor yang mempengaruhi ketidakakuratan algoritma ANN adalah jumlah lapisan maupun parameter lain seperti fungsi aktivasi yang digunakan (Farizawani, dkk., 2020).

KESIMPULAN DAN SARAN

Berdasarkan hasil pengolahan data, prediksi nilai kecepatan gelombang S pada sumur "S-1" menggunakan pendekatan *supervised machine learning* dengan beberapa algoritma menghasilkan nilai yang cukup mendekati nilai kecepatan gelombang S *measured* dari data *sonic log* baik



Gambar 6
Hasil prediksi dan *crossplot* setelah *tuning* algoritma: (a) KNN; (b) SVR.



Gambar 7
Hasil prediksi dan *crossplot* setelah *tuning* algoritma: (a) *Random Forest*; (b) *Decision Tree*.

sebelum maupun setelah *tuning* pada algoritma *library sklearn*, begitupula dengan algoritma ANN. Sedangkan hasil perbandingan antara kelima algoritma yang digunakan menunjukkan bahwa algoritma terbaik adalah *random forest* dengan nilai parameter *n_estimator* terbaik 10. Dan algoritma kedua yang terbaik adalah *k-nearest neighbor* dengan nilai parameter *n_neighbor* terbaik 5.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada PPPTMGB Lemigas yang telah memberikan akses data sehingga penelitian ini dapat dilakukan dengan baik. Selain itu, penulis juga mengucapkan terima kasih kepada editor atas ulasan yang diberikan terhadap penulisan penelitian ini.

DAFTAR ISTILAH / SINGKATAN

Simbol	Definisi	Satuan
Vp	p-wave velocity	us/ft
Vs	s-wave velocity	us/ft
RMSE	Root Mean Square Error	us/ft
MAE	Mean Absolute Error	us/ft
R2	Coefficient of Determination	
KNN	K-Nearest Neighbor	
SVR	Support Vector Regressor	
ANN	Artificial Neural Network	

KEPUSTAKAAN

- Farizawani, A., Puteh, M., Marina, Y. & Rivaie, A.**, 2020. A Review of Artificial Neural Network Learning Rule Based On Multiple Variant of Conjugate Gradient Approaches. Bandung, Indonesia, IOP Publishing Ltd.
- fchollet**, 2020. Introduction to Keras for Engineers. [Online] Available at: https://keras.io/getting_started/intro_to_keras_for_engineers/ [Accessed 18 Juni 2020].
- Huang, G.-B., Zhu, Q.-Y. & Siew, C.-K.**, 2006. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3), pp. 489-501.
- Keras.io**, 2020. About Keras. [Online] Available at: <https://keras.io/about/> [Accessed 5 Desember 2020].
- Maulana, N., Setiawan, B. & Dewi, C.**, 2019. Implementasi Metode Support Vector Regression (SVR) Dalam Peramalan Penjualan Roti (Studi Kasus: Harum Bakery). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(3), pp. 2986-2995.
- Scikit-learn developers**, 2019. Scikit-learn user guide 0.21.3. Scikit-learn developers. [Online] Available at: https://scikit-learn.org/0.21/_downloads/scikit-learn-docs.pdf [Accessed 10 Juli 2020].
- Somvanshi, M., Chavan, P., Tambade, S. & shinde, S. V.**, 2016. Machine Learning Techniques Using Decision Tree and Support Vector Machine. Pune, India, IEEE, pp. 1-7.
- Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., & Deng, H.**, 2019. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology*, 17(1), pp. 26-40.