Lembaran Publikasi Minyak dan Gas Bumi, Vol. 55. No. 2, Agustus 2021: 75 - 87
PUSAT PENELITIAN DAN PENGEMBANGAN TEKNOLOGI MINYAK DAN GAS BUMI
"LEMIGAS"
Journal Homepage: http://www.journal.lemigas.esdm.go.id
ISSN: 2089-3396 e-ISSN: 2598-0300

Algoritma Komputasi *Machine Learning* untuk Aplikasi Prediksi Nilai *Total Organic Carbon* (TOC)

Sanggeni Gali Wardhana[1)], Henry Julois Pakpahan[2)],
Krisdanyolan Simarmata[2)], Waskito Pranowo[2)], dan Humbang Purba[2)]

[1]Universitas Pertamina
Jl. Teuku Nyak Arief, Kebayoran Lama, Jakarta Selatan
[2]Pusat Penelitian dan Pengembangan Teknologi Minyak dan Gas Bumi "LEMIGAS"
Jl. Ciledug Raya Kav. 109, Cipulir, Kebayoran Lama, Jakarta Selatan 12230

Artikel Info:

Naskah Diterima:
31 Mei 2021

Diterima setelah perbaikan:
4 Agustus 2021

Disetujui terbit:
30 Agustus 2021

Kata Kunci:
TOC
machine learning
cross validation
hyperparameter tuning

ABSTRAK

Total Organic Carbon (TOC) merupakan salah satu parameter penting yang digunakan untuk mengevaluasi kemampuan *source rock* secara kuantitas. Pada umumnya, data TOC diperoleh melalui *core* yang kemudian dilakukan proses pirolisis *rock-eval* pada setiap perconto. Namun, proses tersebut memerlukan waku yang cukup lama dan biaya yang cukup besar sehingga data yang didapatkan jumlahnya terbatas. Hal ini akan berimplikasi terhadap validitas penyebaran nilai TOC pada tahapan eksplorasi batuan induk unkonvensional. Data yang terbatas dapat diprediksi dengan pendekatan pola karakterisitik data itu sendiri. Penelitian ini dilakukan bertujuan untuk melakukan prediksi nilai TOC dengan menggunakan algoritma *machine learning* yaitu *Artificial Neural Network, K-Nearest Neighbors, Support Vector Regression, Decision Tree,* dan *Random Forest* dengan memanfaatkan data sumur "A" untuk membangun model dari setiap algoritma *machine learning* dan data sumur "B" untuk mengevaluasi model yang telah dibangun berdasarkan data sumur "A". Pengolahan data untuk memprediksi nilai TOC dimulai dari mempersiapkan data pada sumur "A" berdasarkan korelasi yang tinggi pada prediktor dan data output yang akan diprediksi. Selanjutnya dilakukan pembagian atau *splitting datasets* dengan presentase 60% data digunakan untuk melakukan *training* dan 40% data sebagai *test datasets*. Setelah itu, *train datasets* dapat digunakan untuk membangun model algoritma *machine learning.* Kemudian dilakukan *hyperparameter tuning* dan *cross validation* sehingga dapat dihasilkan model algoritma *machine learning* dengan *hyperparameter* tertentu dengan hasil prediksi yang konsisten. Model terbaik diperoleh berdasarkan hasil *cross validation* dengan menggunakan prediktor dari *test datasets* hasil *splitting* sumur "A" dan *test datasets* dari sumur baru "B". Hasil penelitian menunjukan bahwa hasil prediksi TOC terbaik pada data sumur "A" diperoleh dengan menggunaan algoritma *Random Forest* dan pada sumur "B" menggunakan algoritma K-*Nearest Neighbors*.

PENDAHULUAN

Total Organic Carbon (TOC) adalah ukuran jumlah material organik pada suatu batuan baik kerogen ataupun bitumennya (Peters & Cassa, 1994). Semakin besar nilai TOC maka suatu batuan akan semakin kaya akan material organik yang memungkinkan untuk suatu batuan menjadi penghasil hidrokarbon. Oleh karena itu, nilai TOC penting untuk diketahui guna menilai persebaran serta kualitas dari *source rock* dalam fase eksplorasi hidrokarbon. Pada metode konvensional, TOC umumnya diukur dengan pirolisis *rock-eval* yang dilakukan pada data *core* dan *cutting*, sehingga dalam prosesnya dapat memakan biaya dan waktu yang cukup besar. Selain itu, untuk memprediksi data TOC beberapa studi juga telah dilakukan dengan menggunakan hubungan matematis dan rumus empiris namun sering menghasilkan estimasi yang

Korespondensi:
E-mail: sanggeniwardhana@gmail.com (Sanggeni G.W.)

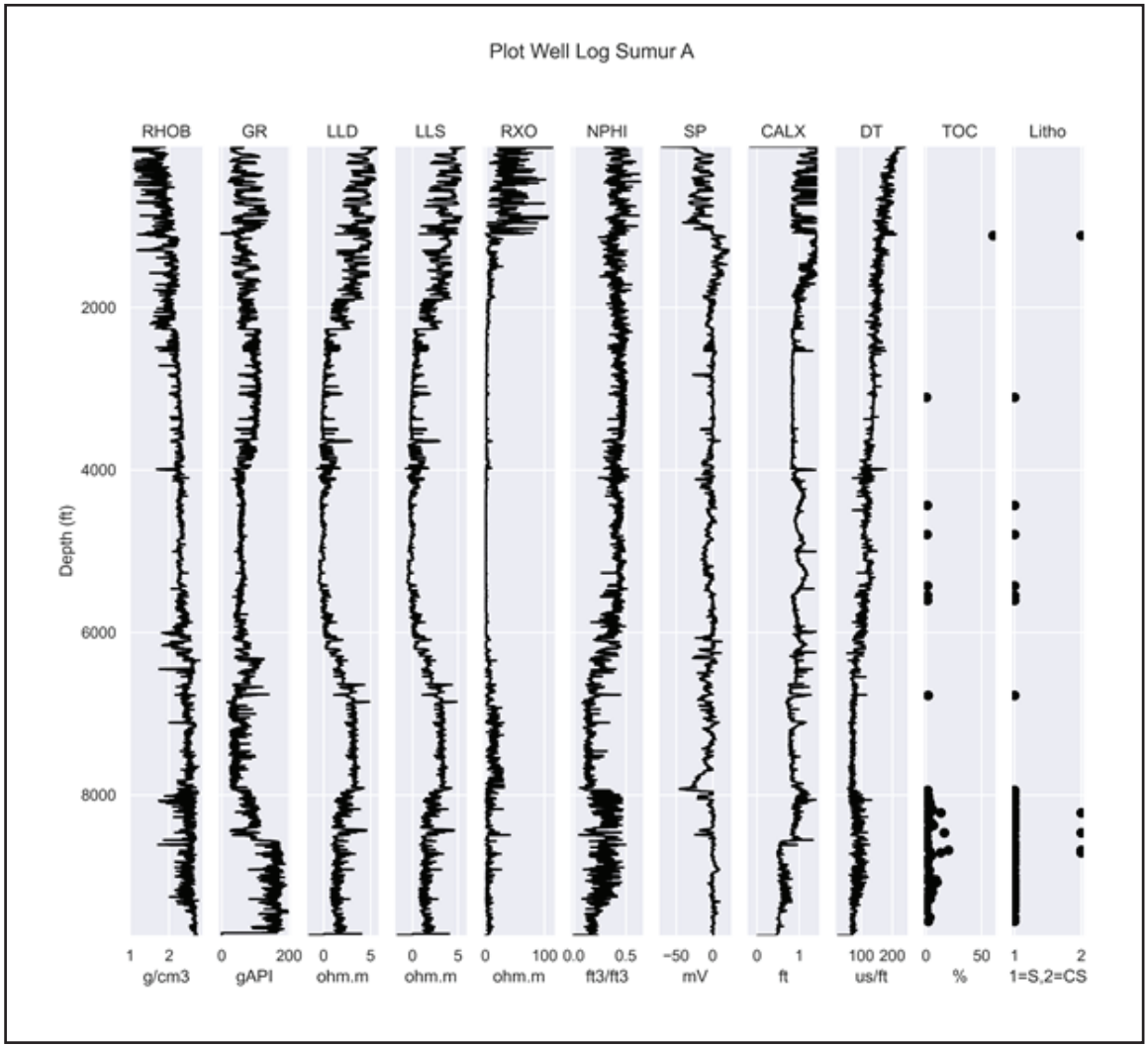
tidak akurat. Karena alasan tersebut metode *machine learning* adalah sebuah alternatif untuk memprediksi dengan data TOC berdasarkan *datasets* yang telah ada berupa perconto TOC dari data *core* serta data sumur tanpa melalui proses analisa melalui data *core* pada seluruh sumur.

Penelitian ini bertujuan untuk melakukan prediksi parameter TOC dengan menggunakan algoritma komputasi *machine learning*, serta melakukan analisa terhadap hasil prediksi dari setiap algoritma *machine learning* yang digunakan.

BAHAN DAN METODE

Data pada penelitian ini menggunakan data sumur yang berada pada cekungan Sumatera Tengah yang terdiri dari atas data sumur dengan informasi tambahan berupa data perconto TOC dan data litologi berdasarkan *core* dengan jumlah 80 perconto yaitu 60 perconto pada sumur "A" dan 20 perconto pada sumur "B". Data sumur "A" digunakan untuk melakukan pelatihan, *test*, serta validasi model dari *machine learning*. Data sumur "B" akan digunakan untuk *blind test* terhadap model yang telah dihasilkan dari pelatihan yang telah dilakukan pada sumur "A".

Pada sumur "A" terdapat beberapa data parameter *well log* yang tersedia (Gambar 1), antara lain adalah data *Depth, Caliper* (CALX), *Density* (RHOB), *Gamma Ray* (GR), *Deep Resistivity* (LLD), *Shallow Resistivity* (LLS), *Resistivity of Flushed Zone* (RXO), *Neutron Porosity* (NPHI), *Self Potential* (SP), *Sonic* (DT). Selain itu, pada setiap perconto *core* data TOC (Gambar 1) terdapat data litologi yang juga dapat digunakan sebagai prediktor pada algoritma *machine learning*.



Gambar 1
Plot data *Well-Log* sumur "A".

Pada penelitian ini data litologi (Gambar 1) yang didapatkan melalui data *core* terdapat dua jenis yaitu litologi *shale* ditunjukan oleh angka "1" sedangkan litologi *coaly shale* ditunjukan oleh angka "2".

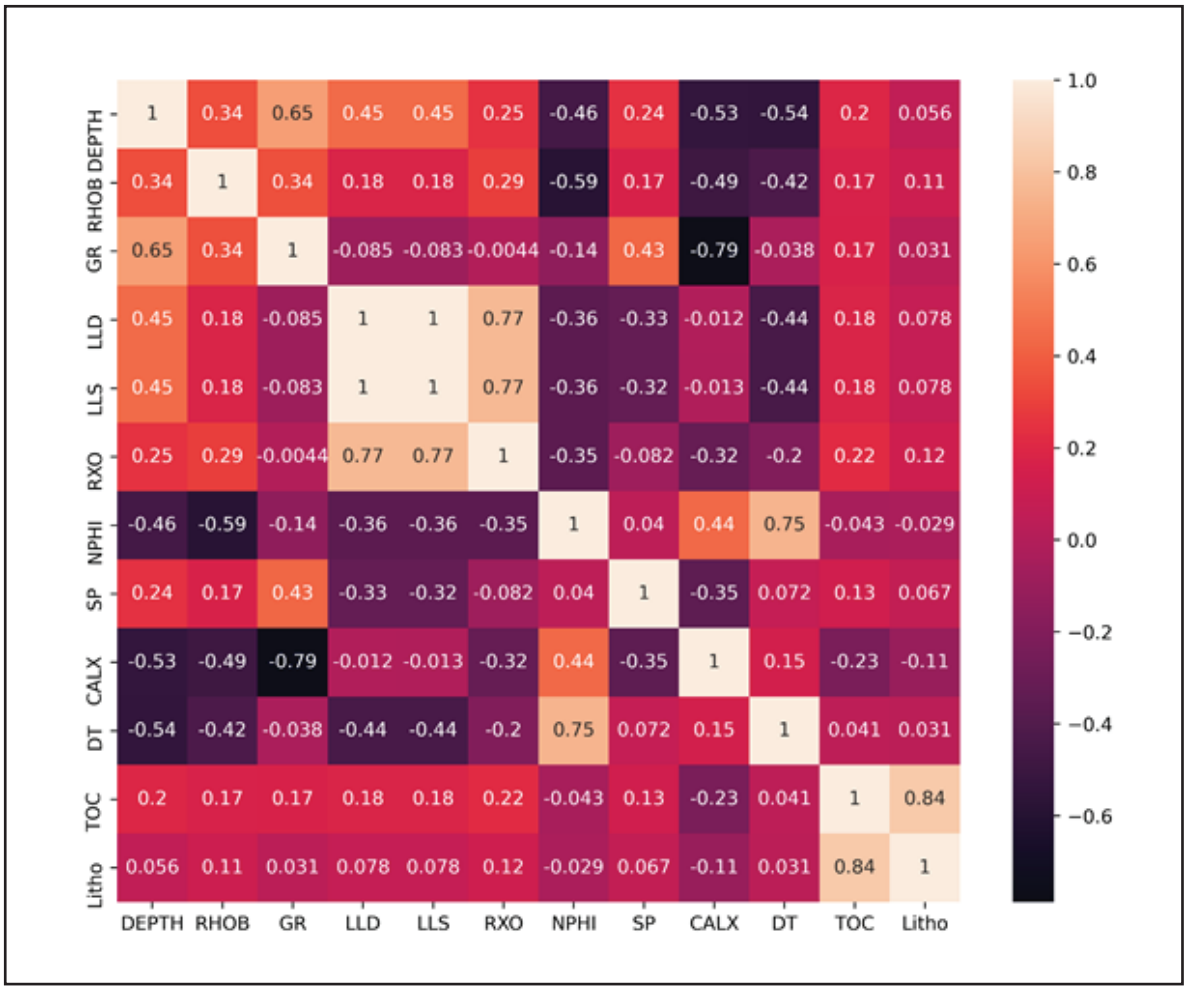
A. Koefisien Korelasi

Seperti yang diketahui, algoritma *machine learning* memanfaatkan data prediktor dan data yang akan diprediksi atau data *output*. Kedua data tersebut akan sangat mempengaruhi proses *training* data untuk membangun model *machine learning* sehingga dapat digunakan untuk memprediksi suatu parameter.

Berdasarkan heatmap (Gambar 2) dapat dilihat bahwa koefisien korelasi dari data TOC dengan semua data sumur menunjukan nilai koefisien korelasi yang cukup rendah. Atas dasar tersebut pemilihan fitur berdasarkan pada Wang (2019) menyatakan bahwa data sumur yang sensitif terhadap nilai TOC adalah *Gamma Ray*, RHOB, LLD, LLS, dan NPHI. Selain itu juga dilakukan penambahan prediktor dengan korelasi yang paling tinggi dengan data TOC untuk membantu dalam proses *training* dengan menggunakan data kedalaman dan data litologi yang didapatkan melalui data perconto *core*. Berikut adalah ringkasan mengenai prediktor (Tabel 1).

Tabel 1
Koefisien korelasi antara
Prediktor dengan data TOC pada Sumur "A"

Prediktor	Korelasi
Lithologi	0.84
Depth	0.2
LLD	0.18
LLS	0.18
GR	0.17
RHOB	0.17
NPHI	-0.043

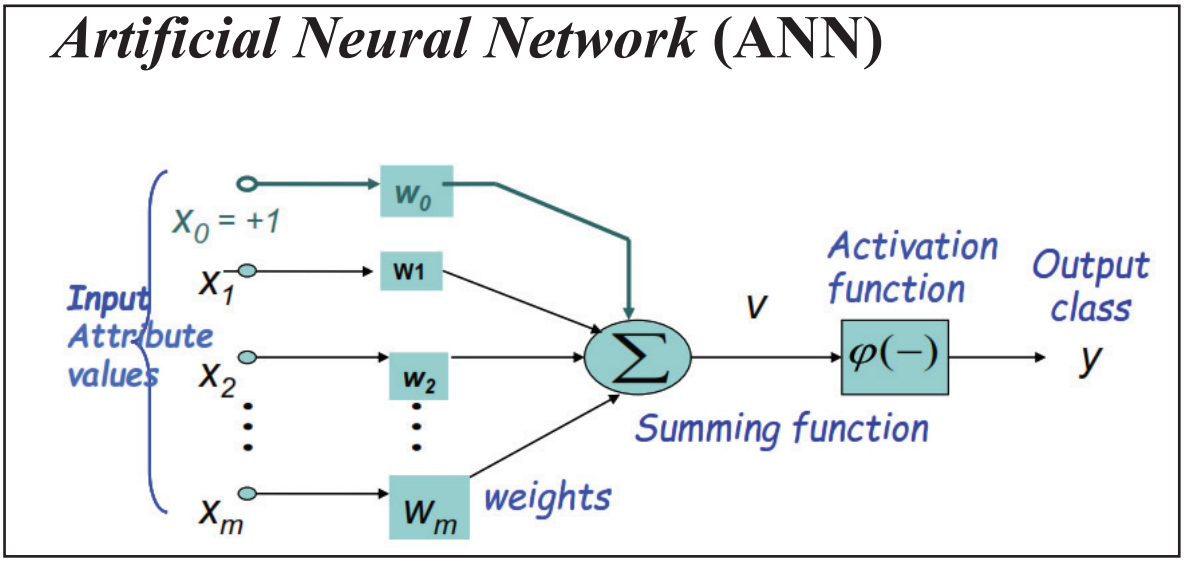


Gambar 2
Heatmap dari seluruh data sumur "A" (koefisien korelasi).

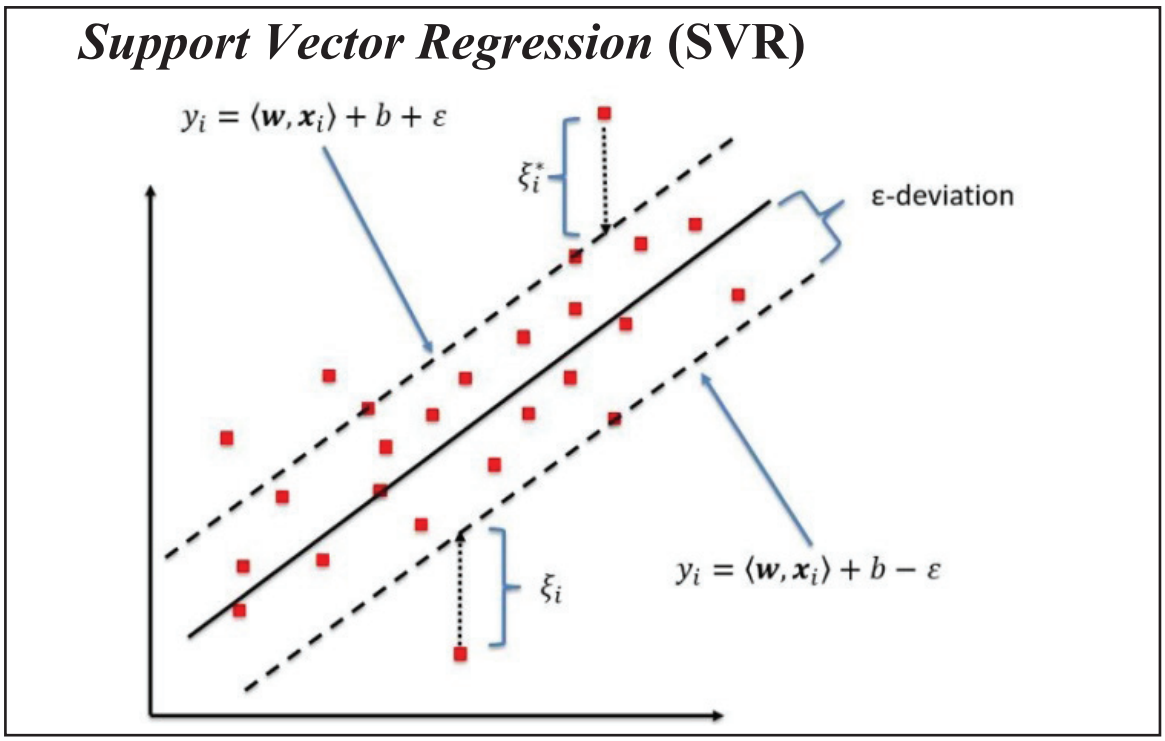
ANN adalah sistem komputasi yang terdiri atas banyak elemen komputasi sederhana yang terintegrasi dalam suatu koneksi yang memiliki bobot tertentu (Isiyaka, dkk., 2019). ANN menirukan bagaimana suatu data dapat disintesis oleh sistem jaringan syaraf biologis. Arsitektur dari ANN didasarkan pada kumpulan dari nodes yang dihubungkan oleh suatu sinapsis. Dalam ANN "sinyal" pada suatu koneksi adalah angka real, dan beberapa fungsi penjumlahan input non-linear menghitung output dari setiap neuron yang dimana pada setiap neuron dan sinapsis memiliki suatu bobot seperti pada (Gambar 3). Untuk menerapkan algoritma ANN, arsitektur jaringan, pembobotan, *learning rate*, dan algoritma *training* harus dipilih secara optimal (Wang, dkk., 2016)

B. *Decision Tree*

Decision tree atau disebut juga pohon keputusan adalah sebuah diagram alur yang memiliki bentuk struktur pohon yang telah dilakukan pengujian



Gambar 3
Ilustrasi arsitektur *neural network* (Kumar, 2003).



Gambar 4
Ilustrasi SVR (Kleynhans T, dkk., 2017).

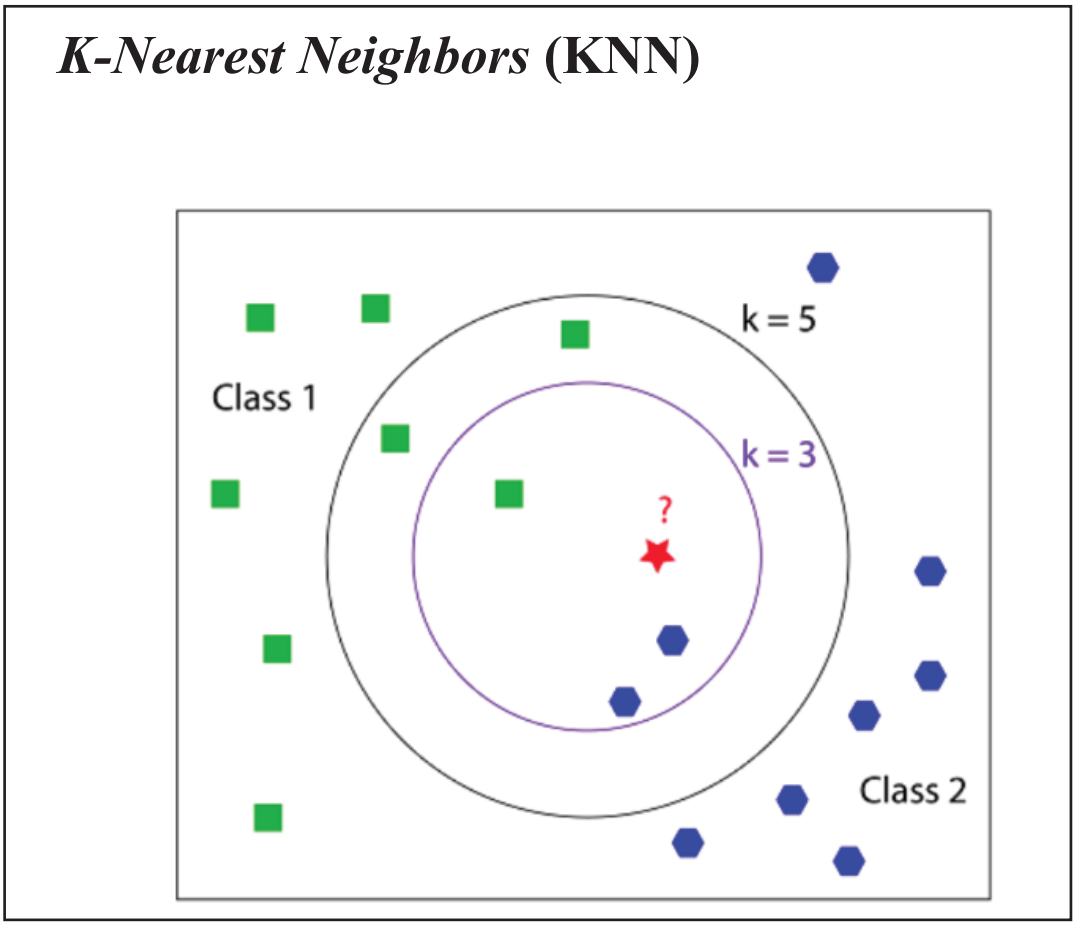
terhadap stautu atribut pada *internal node*, kemudian terdapat cabang yang menyatakan output dari *node*, serta terdapat *leaf node* yang menunjukan distribusi kelas. Tidak hanya digunakan untuk permasalahan klasifikasi, algoritma ini juga dapat digunakan dalam permasalahan regresi.

C. *Random Forest*

Random forest adalah kombinasi dari beragam *tree* yang kemudian dikombinasikan ke dalam suatu model, hal ini dilakukan bertujuan untuk mereduksi variansi dan membuat model lebih baik secara keseluruhannya (Breiman, 2001). Algoritma ini mempunyai kemampuan untuk mengestimasi *feature importance* (fitur penting) dengan cara mengevaluasi seberapa banyak error prediksi jika nilai OOB (*out of bag*) dipermutasikan untuk salah satu fitur ketika fitur lain diubah (Catani, dkk., 2013). Pada algoritma ini dua parameter yang dapat dioptimisasi adalah jumlah dari input fitur tiap nodes dan jumlah dari *regression trees*.

Support Vector Regression (SVR) merupakan pengembangan SVM untuk kasus regresi. Tujuan dari SVR adalah untuk menemukan sebuah fungsi F(x) sebagai suatu *hyperplane* (garis pemisah) berupa fungsi regresi yang sesuai dengan semua input data dengan mengusahakan *error* sekecil mungkin (Schölkopf & Smola, 2002). Dalam SVR, terdapat tiga *hyperparameter* penting yang harus dilakukan optimisasi yaitu nilai constraint violation (C), epsilon (ε), dan gamma (γ). Nilai C menunjukan *tradeoff* (pengorbanan) kompleksitas dari aturan pengambilan keputusan dan tingkat *error* yang terjadi (Cortes & Vapnik, 1995). Epsilon menunjukan efek *smoothness* dari SVM dan kemampuan generalisasi dan kompleksitas jaringan, dan gamma berhubungan dengan permasalahan *underfitting* dan *overfitting*.

KNN adalah algoritma jenis *supervised learning* yang dapat digunakan untuk masalah klasifikasi dan regresi. Seperti pada (Gambar 5), KNN memprediksi sampel baru dari *training* set menggunakan sampel K-terdekat (Kuhn & Johnson, 2013). Algoritma ini baik untuk data *training* yang *noisy* dan cukup berhasil ketika kumpulan data *training* yang besar (Mitchell, 1997). Keuntungan utama KNN adalah



Gambar 5
Ilustrasi KNN (Mitchell, 1997).

algoritma yang sederhana dan asumsi parametrik yang sedikit (Shmueli, dkk., 2016).

D. Alur Kerja Prediksi TOC dengan Algoritma *Machine Learning*

Seperti yang digambarkan pada (Gambar 6), algoritma machine learning dimulai dari input data prediktor dan data output yang akan diprediksi, data prediktor yang dipilih untuk melakukan prediksi adalah yang memiliki korelasi yang tinggi dengan data yang diprediksi yakni data TOC. Setelah itu dilakukan *feature scaling* yang berfungsi untuk melakukan normalisasi terhadap data agar data memiliki skala yang sesuai sehingga dapat meminimalisir kesalahan dalam memprediksi data (Alasadi & Bhaya, 2017). Kemudian dilakukan pembagian atau *splitting datasets* dengan presentase 60% data digunakan untuk melakukan *training* dan 40% data sebagai *test datasets*. Setelah itu *train datasets* dapat digunakan untuk membangun model algoritma *machine learning.* Kemudian akan dilakukan *Hyperparameter Tuning* dan *Cross Validation* dengan menggunakan *Gridsearch Cross Validation.*

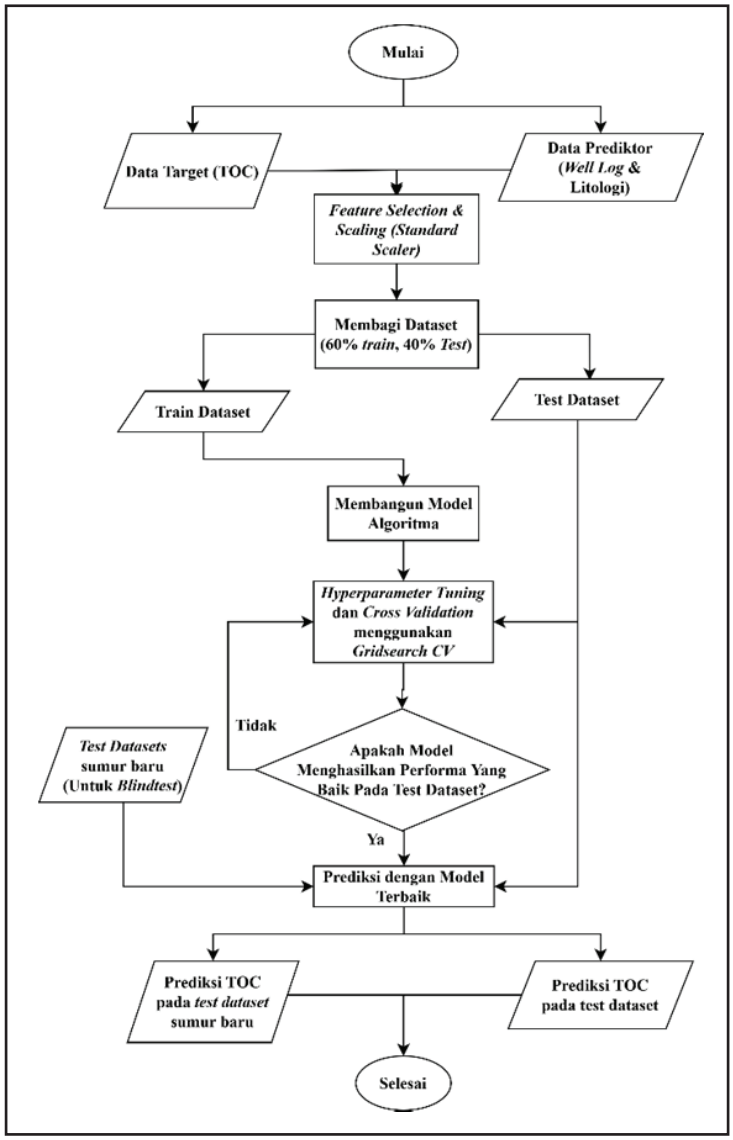
Langkah ini berguna untuk memilih *hyperparameter* terbaik dari setiap algoritma berdasarkan percobaan secara iteratif untuk pada range nilai tertentu. selain itu, *cross validation* juga dilakukan pada langkah ini dengan cara membagi data menjadi beberapa *fold* untuk dicari nilai rata-rata dari akurasi yang dihasilkan dari prediksi pada setiap *fold* untuk memastikan apakah model yang dibangun sudah berjalan secara konsisten.

Langkah selanjutnya adalah menentukan model terbaik berdasarkan hasil *cross validation* yang kemudian digunakan untuk melakukan prediksi dengan menggunakan prediktor dari *test datasets* hasil *splitting* sumur "A" dan *test datasets* dari sumur baru "B" untuk melihat performa model yang dibangun jika dilakukan pada data sumur lain. Setelah itu hasil prediksi nilai TOC dapat dihasilkan.

Pada penelitian ini, untuk algoritma *machine learning Random Forest, Decision Tree*, SVR, dan KNN menggunakan alur kerja pada (Gambar 6). Sedangkan, algoritma ANN memiliki alur kerja yang mirip, namun tidak melalui langkah *hyperparameter tuning* dan *cross validation* dengan menggunakan *grid search crosss validation* seperti pada (Gambar 6).

HASIL DAN DISKUSI

Hal paling penting dalam mengaplikasikan algoritma *machine learning* adalah langkah optimisasi hyperparemeter. Pada penelitian ini, dilakukan dengan menggunakan *gridsearch cross validation.* Pada tahap ini dilakukan secara sekaligus langkah *hyperparameter tuning* dan validasi silang (*cross-validation*) dengan menggunakan *k-fold cross validation* menggunakan penilaian *metrics*



Gambar 6
Flowchart prediksi TOC menggunakan algoritma *decision tree*, *random forest*, SVR, dan KNN.

berupa R^2*score*. Teknis pada proses ini adalah data akan dilatih, lalu model algoritma dari setiap variasi parameter akan divalidasi silang sehingga nilai R^2*score* yang dihasilkan dari setiap variasi parameter pada kurva *train* dan *test* adalah dari hasil yang telah tervalidasi silang.

Pada tahap ini akan dilakukan analisis mengenai parameter model algoritma yang akan digunakan dengan cara menganalisis tingkat kecocokan (*fitting*) dari hasil *training* sehingga dapat menghindari model pelatihan yang menghasilkan prediksi yang *overfitting* maupun *underfitting*.

A. Hasil *Decision Tree*

Performa *Decision Tree* pada (Gambar 7) menunjukan bahwa model mengalami *overfitting* pada nilai *max_depth* yang tinggi, yang berarti model dapat dipelajari pada data *training* dengan baik namun gagal dalam melakukan generalisasi untuk memprediksi data *test*. Pada algoritma ini dipilih skor *test* terbaik yaitu pada saat *max_depth* =1.

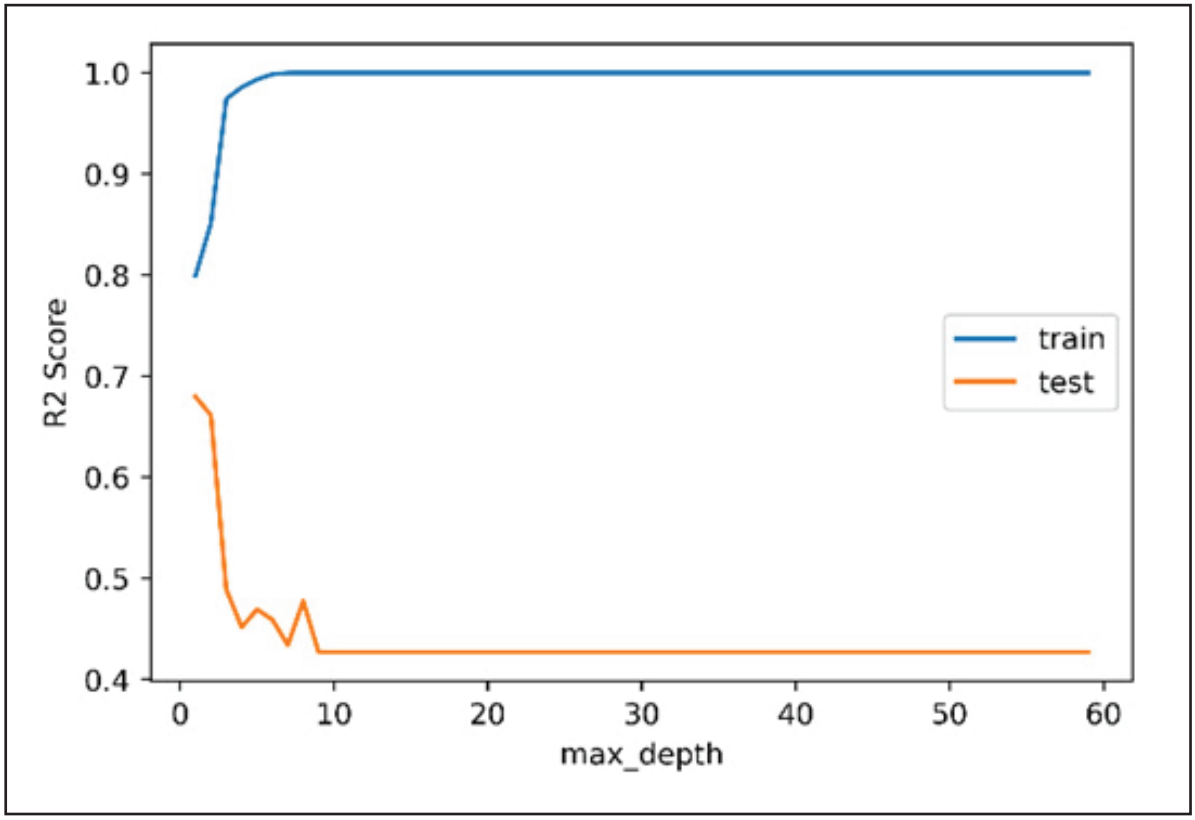
B. Hasil *Random Forest*

Pada algortima *Random Forest* (RF) pada (Gambar 8) kita dapat memberhentikan dan memilih pada saat nilai *n_estimator* = 19, karena peningkatan nilai *n_estimator* dapat menurunkan skor dari *performance test*. Pada kurva ini pada seluruh variasi masih menunjukan *overfitting*, namun tetap dipilih pada saat skor *test* tertinggi karena parameter tersebut adalah yang paling optimal.

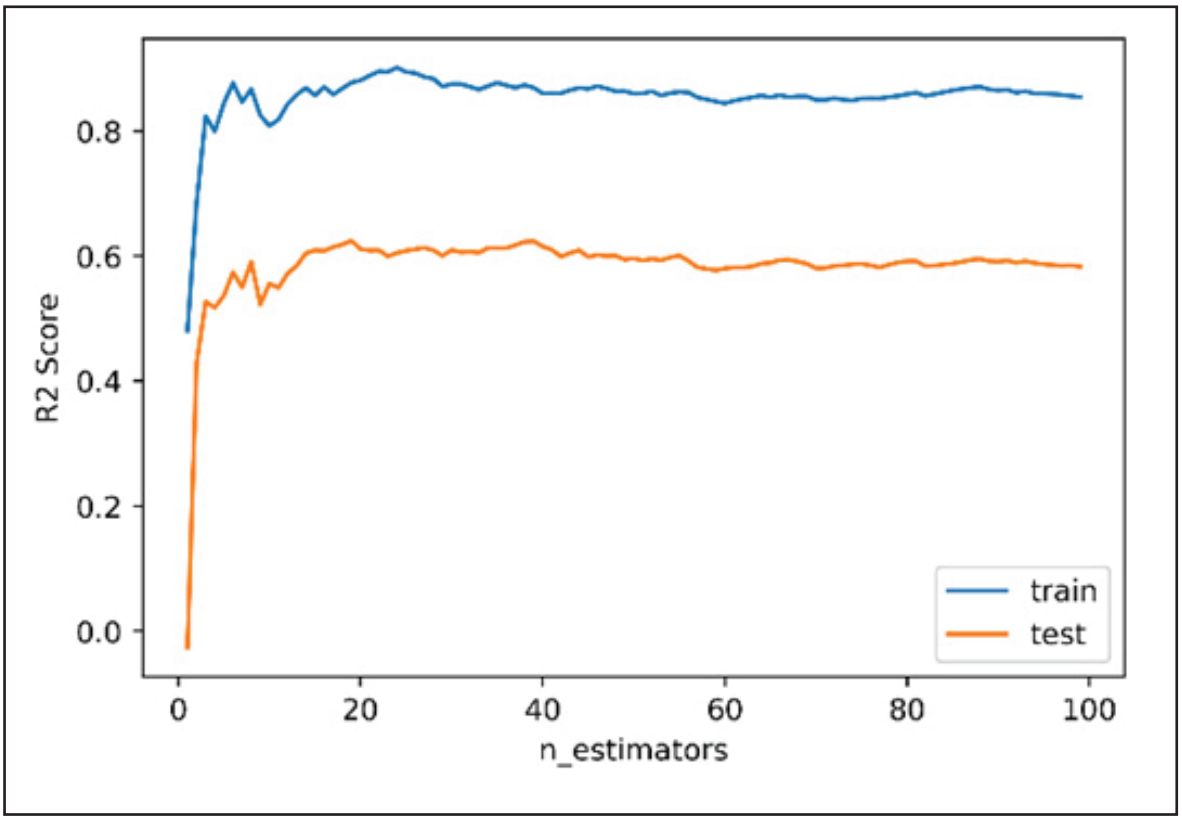
Pada algoritma SVR dalam (Gambar 9) dapat terlihat bahwa model akan semakin *overfitting* pada nilai koefisien C yang semakin tinggi, sehingga *tuning hyperparameter* dapat diberhentikan pada saat skor *test* tertinggi yaitu pada saat nilai C=17. Pada kurva ini pada seluruh variasi masih menunjukan *overfitting*, namun tetap dipilih pada saat skor test tertinggi karena parameter tersebut adalah yang paling optimal.

C. Hasil KNN

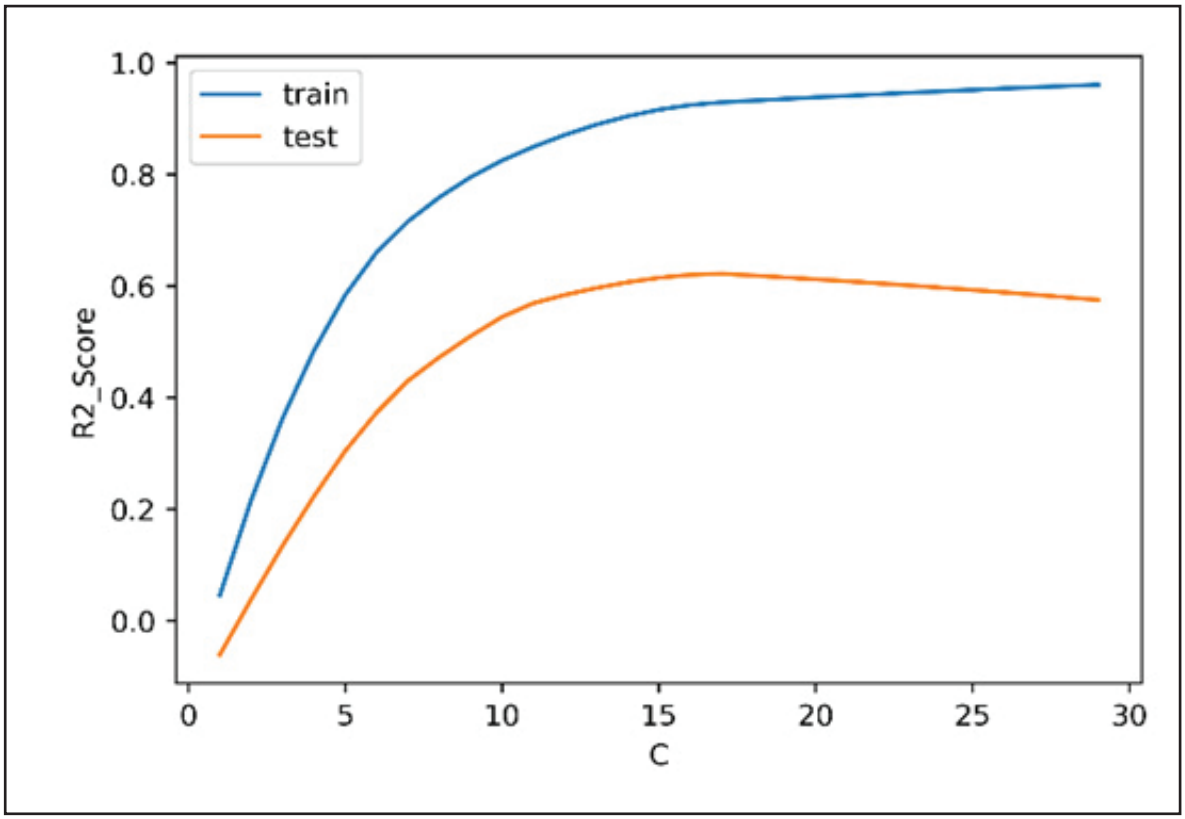
Pada algoritma KNN, saat proses *hyperparameter tuning* pada (Gambar 10) dapat terlihat bahwa saat *n_neighbors*=1 menunjukan *score* 1 yang dimana setiap perconto akan menggunakan nilainya sendiri sebagai referensi dalam proses *training* atau dapat disebut *overfitting*. Pada kasus ini, penambahan *n_neighbors* akan dapat menurunkan skor dari *train* & *test*.



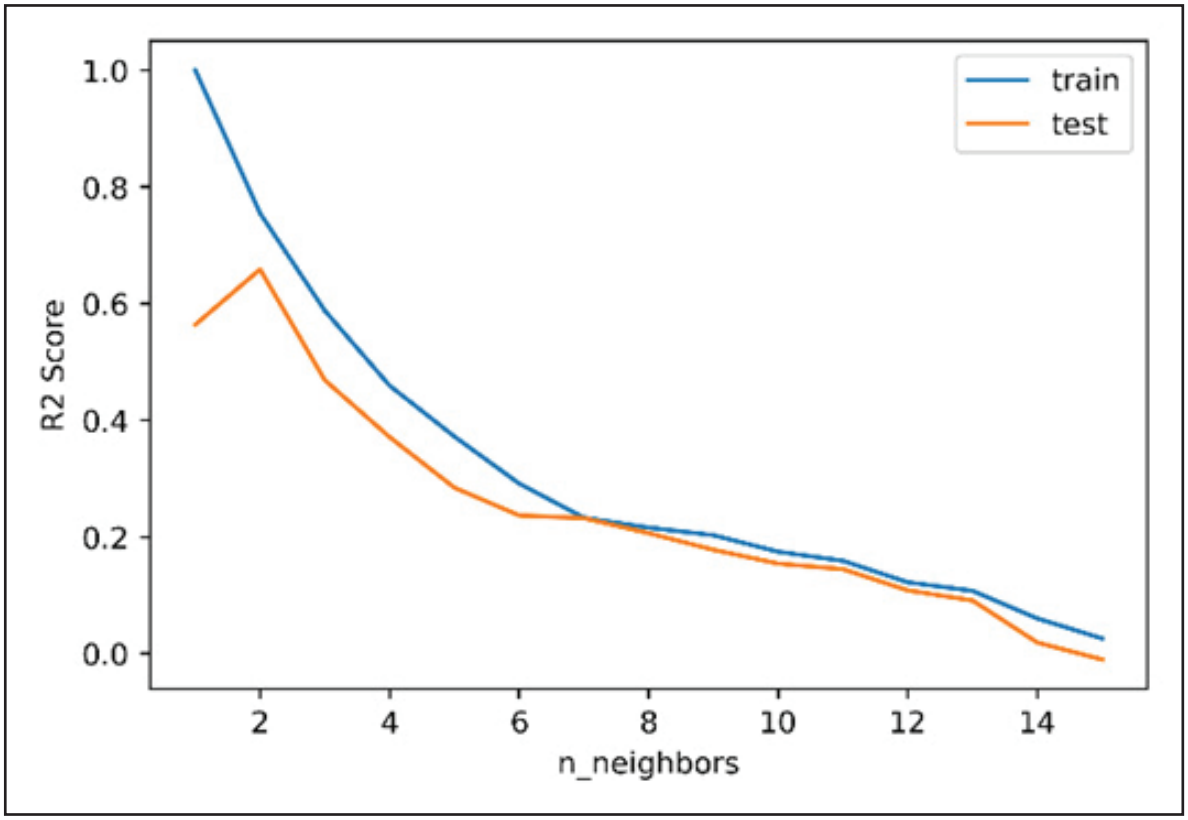
Gambar 7
Performa *decision tree* berdasarkan *max_depth*.



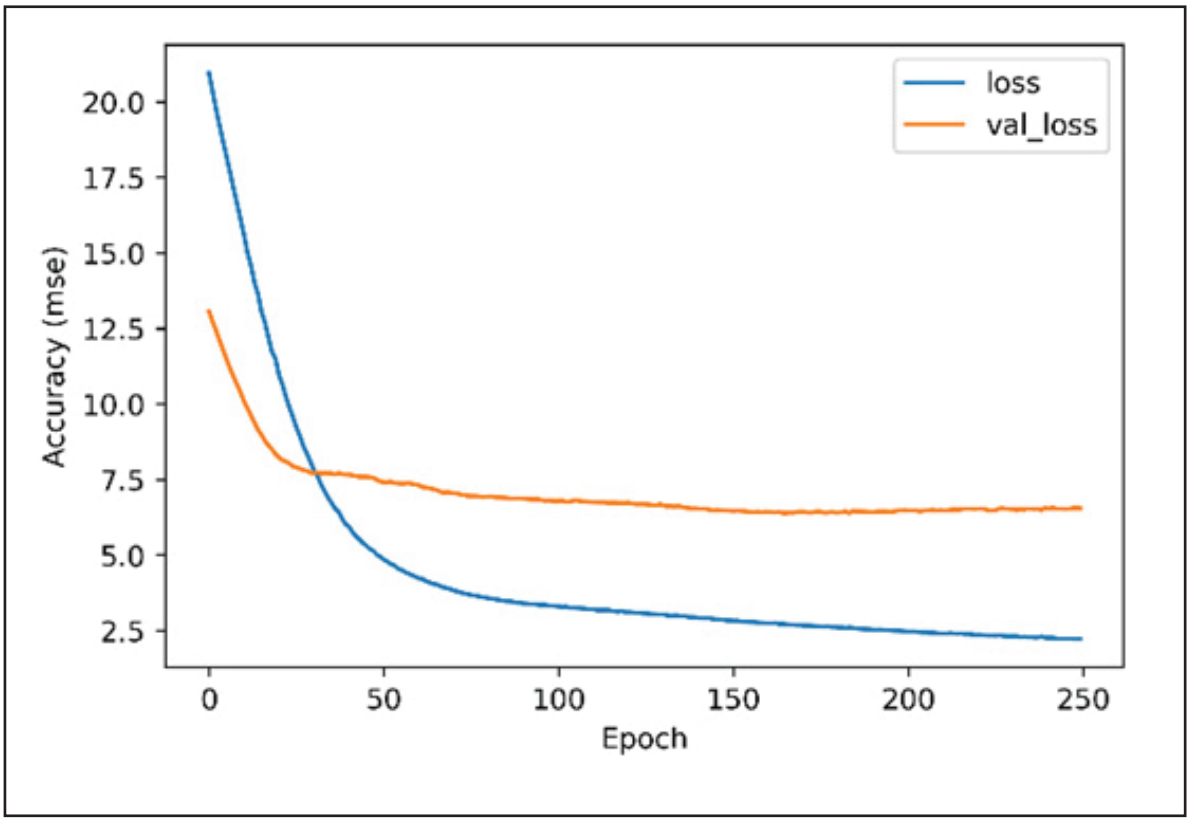
Gambar 8
Performa *random forest* berdasarkan *n-estimator*.



Gambar 9
Performa *support vector regression* berdasarkan koefisien C.



Gambar 10
Performa KNN berdasarkan *n_neighbors* (K).



Gambar 11
Hasil *model loss* pada algoritma ANN.

D. Hasil ANN

Pada algoritma *Artificial Neural Network* (ANN) dapat terlihat bahwa pada *epoch* awal model masih mengalami *underfitting* lalu akan semakin menuju *good fit* pada *epoch* 100, lalu pada *epoch* tinggi model akan semakin mengalami *overfitting* (Gambar 11).

E. Perbandingan Hasil dari Setiap Algoritma

Berdasarkan (Tabel 2) dapat terlihat bahwa hasil prediksi menggunakan pada data *training* menunjukan hasil yang cukup baik. Namun model tersebut jika digunakan pada data *test* tidak menunujukan korelasi yang baik. Hasil *test* tersebut kemudian dilakukan kalkulasi *error* dan skor akurasi yang ditunjukan pada tabel (Tabel 2), hasil tersebut masih menunjukan kondisi *overfitting* pada keseluruhan model algoritma yang ditunjukan oleh skor akurasi *test* lebih rendah daripada skor akurasi *training*.

Pada hasil *test* dataset didapat bahwa algoritma yang menunjukan performa paling baik adalah algoritma *Random Forest dengan* R^2 *Score* = 0.431, diikuti oleh algoritma ANN *dengan* R^2 *Score* = 0.38 dan yang terburuk adalah algoritma *SVR* dengan R^2 *Score* = 0.31. Meskipun masih terjadi *overfitting* pada setiap algoritma, namun model dari hasil *training* ini tetap dipilih, karena *hyperparameter* yang ditentukan adalah yang terbaik dan telah melewati *tuning* dan validasi silang.

Setelah dilakukan pada data *train* dan *test,* maka langkah selanjutnya adalah melakukan prediksi pada data sumur baru yaitu sumur "B" sehingga dapat dihasilkan prediksi TOC pada sumur "B".

Tabel 3 diatas menunjukan performa dari setiap model algoritma *machine learning* yang telah diaplikasikan untuk prediksi TOC pada sumur lain yaitu sumur "B". Hasil ini diharapkan dapat menunjukan gambaran mengenai bagaimana jika suatu model yang dibuat pada suatu data sumur kemudian diaplikasikan pada data sumur lain.

Berdasarkan hasil (Tabel 3) tersebut, algoritma yang memiliki R^2*score* tertinggi dan error terendah adalah algoritma KNN, hal ini tidak sesuai dengan Tabel 2 karena terdapat kemungkinan terjadinya *overfitting* pada saat proses *training* data sehingga menyebabkan algoritma *Random Forest* menempati

Tabel 2
Perbandingan skor performa fase *train* dan *Test* dari
setiap model berdasarkan MAE, MSE, RMSE, dan R^2 *Score* (Sumur A)

	KNN		SVR		DT		Random Forest		ANN	
	Training	Test	Training	Test	Training	Test	Training	Test	Training	Test
MAE	1.058	1.65	0.682	2.05	1.469	1.86	2.223	2.82	0.748	2
MSE	1.86	6.48	1.94	6.9	4.478	6.31	2.774	5.65	0.911	6.18
RMSE	1.364	2.55	1.394	2.63	2.116	2.51	1.665	2.38	0.954	2.49
R2	0.906	0.35	0.902	0.31	0.775	0.365	0.86	0.431	0.954	0.38

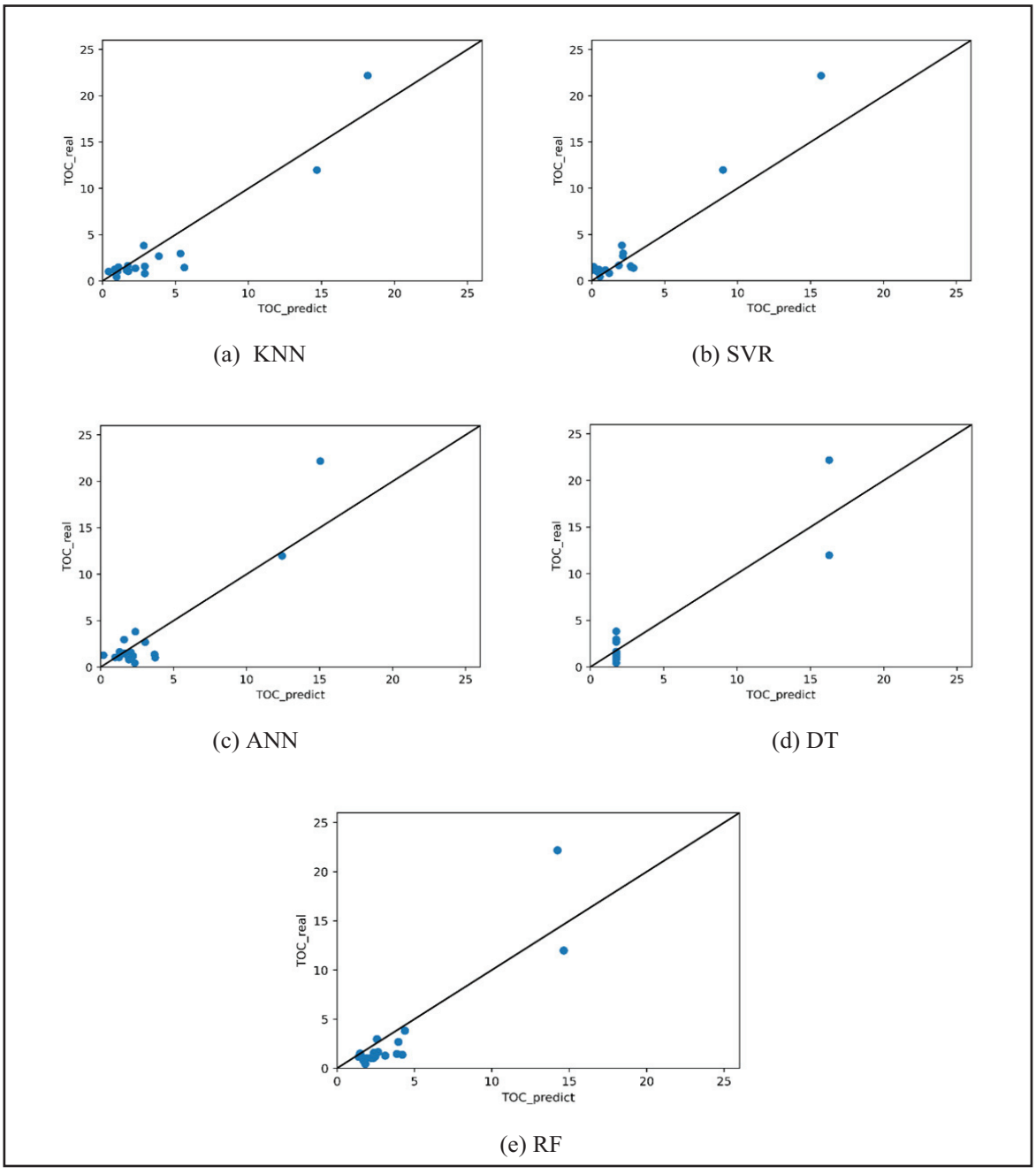
Tabel 3
Skor performa dari setiap algoritma berdasarkan MAE, MSE, RMSE, dan R^2 *Score* (Sumur B)

Metrics	KNN	SVR	DT	RF	ANN
MAE	1.239	1.301	1.163	1.52	1.178
MSE	3.017	3.63	3.453	5.206	3.428
RMSE	1.737	1.905	1.858	2.281	1.851
R2	0.884	0.86	0.867	0.8	0.868

urutan pertama pada fase *training* dan *test* dalam Tabel 2. Hasil prediksi TOC dengan menggunakan KNN menunjukan korelasi yang cukup baik antara nilai TOC prediksi dan nilai TOC observasi pada Gambar 13(a), model ini dapat memprediksi dengan cukup baik pada nilai TOC rendah namun kurang baik pada saat nilai TOC tinggi.

Kemudian algoritma *machine learning* yang menempati urutan selanjutnya adalah algoritma SVR dan ANN, pada Gambar 13 (b) dan (c) dapat terlihat bahwa hasil prediksi dari algoritma SVR dan ANN menunjukan *crossplot* yang mirip, dengan korelasi yang cukup baik pada TOC rendah namun kurang baik pada nilai TOC tinggi.

Lalu pada urutan keempat adalah algoritma *Decision Tree* dengan R^2 *Score* sebesar 0.867, namun hasil model ini menunjukan hasil prediksi nilai TOC yang tidak variatif seperti pada Gambar 13(d), sehingga model ini tidak direkomendasikan untuk digunakan pada kasus ini.

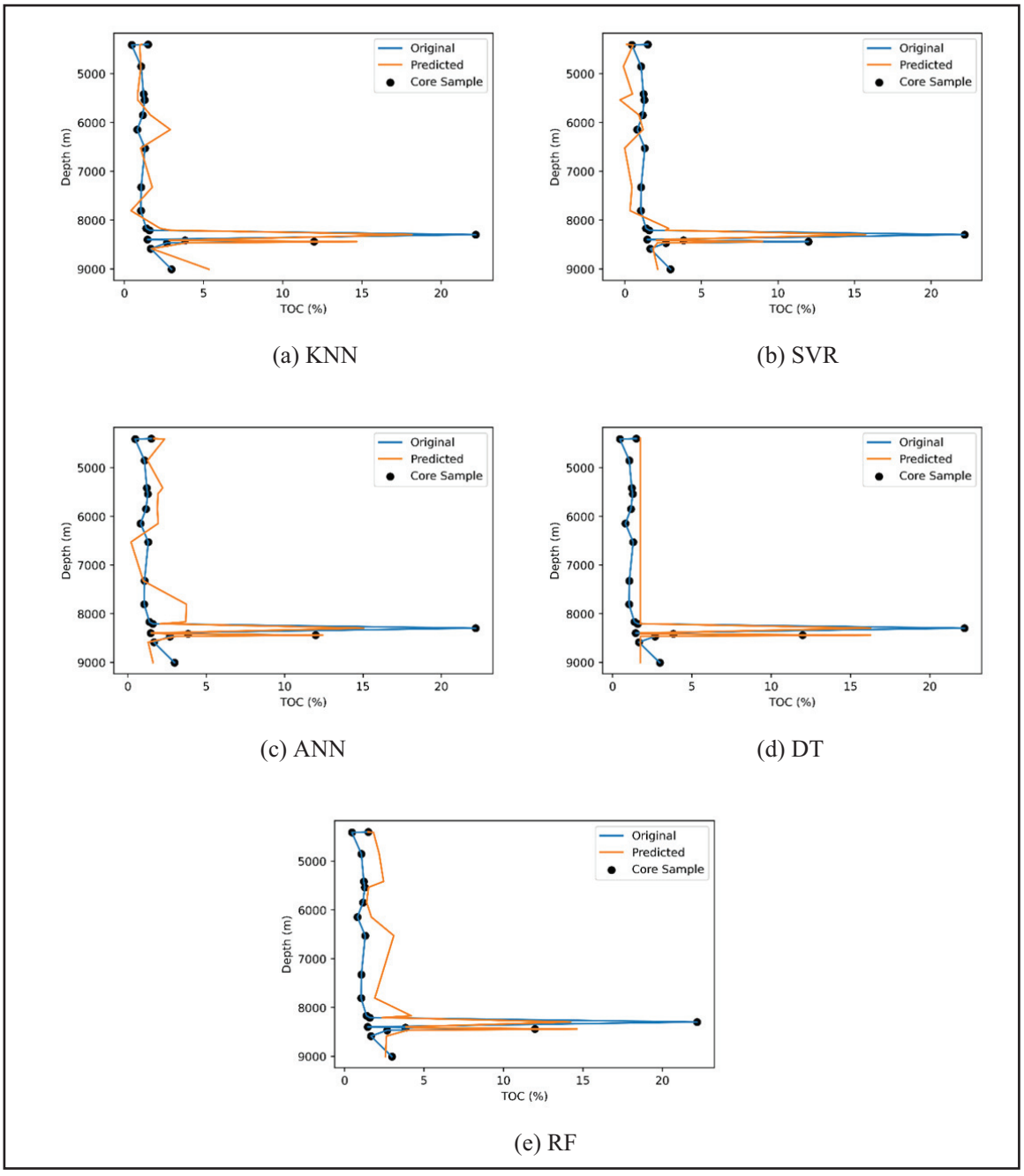


Gambar 12
Crossplot TOC observasi dengan TOC hasil prediksi
pada sumur "B" : (a) KNN, (b) SVR, (c) ANN, (d) DT, (e) RF.

Hasil dengan R^2*score* terendah dan error tertinggi adalah pada algoritma *Random Forest* dengan nilai R^2*score* sebesar 0.800. Pada model ini, nilai TOC rendah tidak dapat diprediksi dengan baik Gambar 13(e), sehingga membuat skor yang dihasilkan lebih rendah daripada algoritma lainnya.

KESIMPULAN DAN SARAN

Berdasarkan percobaan yang dilakukan pada penelitian ini, prediksi nilai *Total Organic Carbon* (TOC) dengan menggunakan algoritma *machine learning* KNN, SVR, RF, DT, dan ANN dengan menggunakan input berupa data GR, LLD, LLS, NPHI, dan RHOB, dan Depth serta tambahan data lithologi pada data *core* cukup berhasil dilakukan pada sumur "A" dan "B". Pada sumur "A" sebagai data *training* didapatkan model terbaik ditunjukan oleh model ada algoritma RF dengan nilai R^2*score* sebesar 0.431, diikuti dengan algoritma ANN, DT, dan KNN, sedangkan hasil terburuk ditunjukan oleh algoritma SVR dengan dengan nilai R^2*score* sebesar 0.31. Pada sumur "B" sebagai data untuk melakukan



Gambar 13
Hasil prediksi TOC pada Sumur "B" : (a) KNN, (b) SVR, (c) ANN, (d) DT, (e) RF.

validasi jika dilakukan *blindtest* pada sumur selain data *training*, dihasilkan algoritma terbaik adalah algoritma KNN dengan nilai R^2*score* sebesar 0.884 dan algoritma dengan hasil paling buruk adalah *Random Forest* dengan nilai R^2*score* sebesar 0.800.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada LEMIGAS atas ketersediaan seluruh *datasets* yang diperlukan pada penelitian ini dan para *reviewer* yang tidak dapat disebutkan satu persatu.

DAFTAR ISTILAH/ SINGKATAN

Simbol	Definisi	Satuan
ANN	*Artificial Neural Network*	
KNN	*K-Nearest Neighbors*	
SVR	*Support Vector Regression*	
DT	*Decision Tree*	
RF	*Random Forest*	
CV	*Cross-Validation*	
MAE	*Mean Absolute Error*	
MSE	*Mean Square Error*	
RMSE	*Root Mean Square Error*	
R^2 *Score*	*R Square Score*	

KEPUSTAKAAN

Alasadi, S. & **Bhaya, W.**, 2017. Review of Data Preprocessing Techniques in Data Mining. *Journal of Engineering and Applied Sciences,* 12(16), pp. 4102-4107.

Breiman, L., 2001. Random Forests. *Machine Learning,* Volume 45, p. 5–32.

Catani, F., Lagomarsino, D., Segoni, S. & **Tofani, V.,** 2013. Landslide Susceptibility Estimation by Random Forests Technique: Sensitivity and Scaling Issues. *Natural Hazards and Earth System Sciences,* 13(11), pp. 2815-2831.

Cortes, C. & **Vapnik, V.,** 1995. Support-Vector Networks. *Machine Learning,* Volume 20, p. 273–297.

Isiyaka, H. A., Mustapha, A., Juahir, H. & **Phil-Eze, P.,** 2019. Water Quality Modelling Using Artificial Neural Network and Multivariate Statistical Techniques. *Modeling Earth Systems and Environment,* Volume 5, p. 583–59.

Kleynhans, T., Montanaro, M., Gerace, A. & **Kanan, C.**, 2017. Predicting Top-of-Atmosphere Thermal Radiance Using MERRA-2 Atmospheric Data with Deep Learning. *Remote Sensing,* 9(11), p. 1133.

Kuhn, M. & **Johnson, K.,** 2013. *Applied Predictive Modeling.* New York: Springer.

Kumar, A., 2003. Neural Network Based Detection of Local Textile Defects. *Pattern Recognition,* 36(7), pp. 1645-1659.

Mitchell, T. M., 1997. *Machine Learning*:McGraw Hill.

Peters, K. & **Cassa, M. R.,** 1994. Applied Source Rock Geochemistry: Chapter 5: Part II. Essential Elements. In: *The Petroleum System--From Source to Trap :*AAPG Special Volumes, pp. 93-120.

Schölkopf, B. & **Smola, A. J.,** 2002. *Learning With Kernels.* Cambridge: MIT Press.

Shmueli, G., Patel, N. R. & **Bruce, P. C.,** 2016. *Data Mining for Business Intelligence: Concepts.* India: Techniques and Applications Wiley.

Wang, H., Wu, W., Chen, T., Dong, X., & **Wang,G**., 2019. An Improved Neural Network for TOC, S1 and S2 Estimation based on Conventional Well Logs. *Journal of Petroleum Science and Engineering,* Volume 176, pp. 664-678.

Wang, L.-J., Guo, M., Sawada, K., Lin, J., & **Zhang, J.**, 2016. A Comparative Study of Landslide Susceptibility Maps using Logistic Regression, Frequency Ratio, Decision Tree, Weights of Evidence and Artificial Neural Network. *Geosciences Journal,* Volume 20, p. 117-136.