Monitoring Kualitas Air Sungai Secara Realtime Berbasis Internet Of Things Dan Big Data

Riyadh Arridha 1[1,a]
[1] Jurusan Manajemen Informatika, Politeknik Negeri Fakfak, Jl. Imam Bonjol Atas, Air Merah, Wagom, Fakfak, 98612, Indonesia
[a] riyadh.arridha@gmail.com

***Abstract*—Monitoring water conditions in real-time is a critical mission to preserve the water ecosystem in maritime and archipelagic countries, such as Indonesia that is relying on the wealth of water resources. To integrate the water monitoring system into the big data technology for real-time analysis, we have engaged in the ongoing project named SEMAR (Smart Environment Monitoring and Analytic in Real-time system), which provides the IoT-Big Data platform for water monitoring. However, SEMAR does not have an analytical system yet. This paper proposes the analytical system for water quality classification using Pollution Index method, which is an extension of SEMAR. Besides, the communication protocol is updated from REST to MQTT. Furthermore, the real-time user interface is implemented for visualisation. The evaluations confirmed that the data analytic function adopting the linear SVM and Decision Tree algorithms achieves more than 90% for the estimation accuracy with 0.019075 for the MSE. The processing time of the SEMAR system only takes an average 0.5 seconds to process the data to be visualized.**

Keywords—SEMAR, water condition monitoring, real-time analysis, IoT, big data, classification, machine learning.

***Abstrak*—Sistem monitoring lingkungan air secara realtime merupakan hal yang harus dilakukan dalam upaya menjaga ekosistem air pada negara maritim dan kepulauan, seperti contoh Indonesia yang memiliki kekayaan sumber daya air yang sangat besar. Dalam rangka mengintegrasikan sistem monitoring dengan teknologi big data untuk analisa realtime, kami mambangun sebuah sistem yang dinamakan SEMAR (Smart Environtment Monitoring and Analyttic in Realtime system), yang merupakan platform IoT-Big data untuk monitoring lingkungan air. Akan tetapi, SEMAR belum memiliki sistem analisa. Sehingga kami melakukan penelitian untuk membangun sistem analisa untuk klasifikasi kualitas air menggunakan metode Indeks Pencemaran, yang mana akan menjadi ekstensi buat sistem SEMAR. Hasil evaluasi menunjukkan algoritma Linear SVM dan Decision Tree yang digunakan memiliki tingkat akurasi di atas 90% dan rata-rata MSE sebesar 0.019075. Sementara untuk waktu pemrosesan sistem SEMAR hanya membutuhkan rata-rata 0.5 detik untuk memproses data yang diterima hingga proses visualisasi.**

Kata Kunci—SEMAR, monitoring kualitas air, analisa realtime, IoT, Big Data, klasifikasi, machine learning

I. Pendahuluan

Indonesia adalah negara maritim dan negara kepulauan yang terbentang lebih dari 17.000 pulau dari Sabang sampai Merauke. Secara persentase luas laut Indonesia adalah 70% dari luas Indonesia secara keseluruhan. Kekayaan laut Indonesia tidak dapat terbantahkan lagi dan posisinya sangat vital untuk menunjang kelangsungan hidup banyak masyarakatnya. Namun hal tersebut tidak dilandasi dengan kesadaran akan pentingnya menjaga kelangsungan ekosistem laut. Eksploitasi yang tidak bertanggung jawab dan cenderung merusak mengakibatkan hanya sekitar 5% dari terumbu karang di Indonesia yang masih terjaga.

Senada dengan lautnya, sebagian besar sungai di Indonesia juga mengalami kerusakan ekosistem akibat pencemaran. Pencemaran tersebut diantaranya bersumber dari limbah industri dan rumah tangga. Kurangnya kesadaran warga, lemahnya pengawasan pemerintah dan keengganan mereka untuk melakukan penegakan hukum yang benar menjadikan masalah pencemaran sungai menjadi hal yang semakin mengkhawatirkan.

Sebuah metode dalam ilmu komputasi dalam menyelesaikan persoalan yang secara matematis tidak dapat diselesaikan dilakukan dengan menggunakan pendekatan kecerdasan buatan. Dengan menggunakan pendekatan kecerdasan buatan maka akan dibuat sebuah model yang dapat menklasifikasi, menklaster, memprediksi suatu data untuk cenderung mengarah ke bagian/arah tertentu. Kemampuan tersebut diperoleh dengan adanya proses pembelajaran terlebih dahulu. Dengan menggunakan data pembelajaran, sebuah system kecerdasan buatan akan dibekali dengan

pengetahuan untuk memprediksi data-data yang mendatang untuk cenderung ke golongan tertentu sehingga mesin dapat memberikan kesimpulan dari hasil pembelajarannya tersebut.

Dalam penelitian ini, diusulkan sebuah integrase sistem antara Internet of Things dan Big data disertai metode untuk mendeteksi kualitas air dengan menggunakan pendekatan kecerdasan buatan. Hal ini diharapkan dapat memberikan solusi yang lebih mudah dan murah kepada pihak-pihak terkait maupun kepada masyarakat dalam mengetahui kualitas air pada suatu badan air.

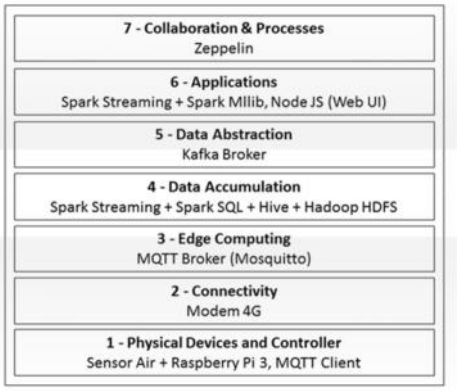
Berikut ini mengenai penelitian terkait yang mencoba untuk menyelesaikan permasalahan dalam penelitian ini antara lain: SEMAR terdiri dari: 1) ROV Water Quality Monitoring System [1], ide tentang ROV Water Quality Monitoring System berdasarkan pada pengalaman Badan Lingkungan pemerintah yang biasanya mengambil sampel untuk memonitor kondisi air sungai. Pada penelitian ini kami mengajukan cara yang berbeda dengan mengkombinasikan Remotely Operated Vehicle (ROV) atau robot air kecil yang dapat dikontrol dengan sensor kualitas air. Dengan alat ini, petugas tidak perlu mengambil sampel air secara manual. Dan hasil dari sensor dapat langsung dikirim ke server melalui internet; 2) Wireless Mesh Network [2], dalam SEMAR kami menggunakan teknologi wireless mesh untuk memperlebar jangkauan komunikasi antara sensor dan server; 3) Portable Water Quality Monitoring System [3], kami mengembangkan sebuah alat portable dan murah untuk pemantauan kualitas air yang hasilnya dapat langsung dikirm ke server; 4) Coral Reef Monitoring System [4], sistem ini dibangun untuk melakukan monitoring terhadap kondisi karang pada perairan dangkal dengan menggunakan kamera aktif dan datanya langsung dikirim ke server; 5) Big Data Storage Architecture [5], data dari sensor dikumpulkan dan disimpan pada Hadoop server dengan dukungan HDFS, Yarn, dan MapReduce. Modaresi et. al. [6], telah melakukan studi mengenai klasifikasi kualitas air menggunakan metode Indeks Kualitas Air CCME (Canadian Council of Minister of the Environment) dengan 2 parameter yaitu Nitrate and Chloride. Dalam penelitian ini menggunakan 3 algoritma yaitu Support Vector Machines, Probabilistic Neural Networks, and K-Nearest Neighbor. Hasil penelitian tersebut menunjukkan algoritma SVM menampilkan best performance dengan tidak ada error pada proses kalibrasi dan validasi. Ladjal et al. [7], juga telah melakukan studi mengenai klasifikasi kualitas air menggunakan Dempster-Shafer Theory. Dalam penelitian ini menggunakan 4 parameter yaitu Temperature, pH, Conductivity, dan Turbidity. Algoritma yang digunakan adalah ANN dan SVM. Hasil dari penelitian ini menunjukkan algoritma SVM memiliki performa yang lebih baik dibandingkan menggunakan ANN. Jaloree et al. [8], juga telah melakukan studi mengenai klasifikasi kualitas air menggunakan algoritma Decision Tree. Parameter yang digunakan dalam penentuan kualitas air adalah pH, DO, BOD, No3_N, dan NH3_N. Dimana hasil dari proses training menunjukkan tingkat akurasi 95.4545%. Saghebian et al. [9] telah melakukan study tentang klasifikasi kualitas air menggunakan Decision Tree. Hasil study menunjukkan Decision Tree mampu memproses dataset yang digunakan. Hasil studi juga menunjukkan rata-rata CCI (Correctly Classified Instances) dan Statistik Kappa untuk prediksi kualitas air berada pada angka 0.88 dan 0.83%. Keempat penelitian di atas dalam pengaplikasiannya belum mendukung proses klasifikasi realtime dan belum terintegrasi dengan teknologi Big Data. Fazio et al. [10], telah melakukan studi mengenai implementasi big data sebagai media penyimpanan untuk Smart Environment Monitoring. Dalam penelitian tersebut digambarkan secara umum mengenai System in Cloud environment for Advanced Multi-risk Management (SIGMA) yang merupakan bagian dari Italian National Operative Program (PON). Project tersebut diharapkan dapat mengakomodir segala macam data yang bersumber dari berbagai environment. Dalam system tersebut belum terlihat gambaran adanya proses Big Data Analytic. Meng et al. [11], telah melakukan studi mengenai performa antara Mahout dan MLLib Spark v1.1 dan v1.4 menggunakan Amazon Reviews dataset yang dijalankan pada 16 node. Hasil penelitian tersebut menunjukkan penjadwalan MapReduce pada Mahout menjadi overhead and lack pada dataset ukuran menengah. Hal yang berbeda pada Spark MLlib yang mengalami kinerja yang sangat baik dan bahkan dapat menangani

dataset dengan ukuran lebih besar. Sementara itu Richter et al. [12], telah melakukan studi komparasi beberapa toolkit untuk machine learning pada Big Data yaitu Mahout MapReduce, Mahout Samsara, MLlib Spark, H2O, dan SAMOA. Dari penelitian tersebut secara ratarata MLlib Spark dan H2O memiliki performa lebih baik dibanding toolkit yang lain dalam hal Extensibility, Scalability, Usability, Fault Tolerance and Speed. Namun dari sisi implementasi algoritma, MLlib Spark merupakan tool dengan implementasi terbanyak dengan total sebanyak 17 algoritma dibandingkan dengan tool yang lain seperti Mahout MapReduce sebanyak 13 algoritma, H2O 10 algoritma, Mahout Samsara 7 algoritma and SAMOA 3 algoritma. Kelebihan lain dari MLlib Spark dibandingkan yang lain adalah kemampuannya yang dapat menangani proses batch maupun stream. Berbeda dengan Mahout dan H2O yang hanya menangani proses batch, dan SAMOA yang hanya menangani proses stream.

II. Metode Penelitian

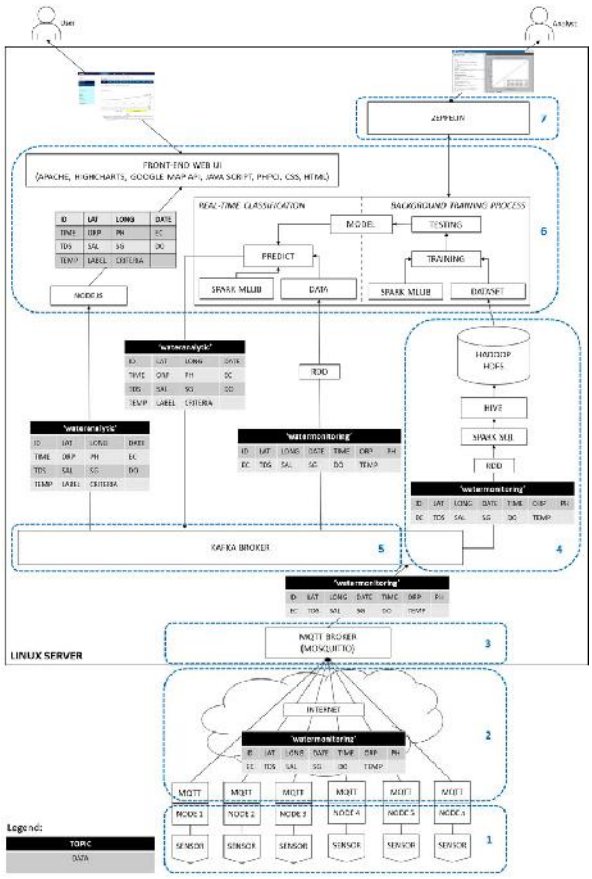
A. Metode

Dalam penelitian ini digunakan teknologi Internet of Things dan Big Data dalam mengembangkan dan mendesain sistem. Secara umum desain sistem didasarkan pada tujuh lapisan pada teknologi Internet of Things yang dapat dilihat pada gambar 1. Pengadopsian teknologi Big Data akan diintegrasikan ke dalam beberapa lapisan dari IoT.



Gambar 1. Gambaran Umum Sistem

Integrasi dari keseluruhan layer akan menghasilkan sebuah platform yang digunakan untuk tujuan tertentu seperti platform untuk monitoring lingkungan, kesehatan, lalu lintas, dll. Dalam penelitian ini akan dihasilkan sebuah platform yang dapat digunakan untuk monitoring lingkungan, utamanya lingkungan air. Gambar 2 menunjukkan detail desain sistem yang dibangun dalam penelitian ini yang didasarkan pada tujuh lapisan Internet of Things.

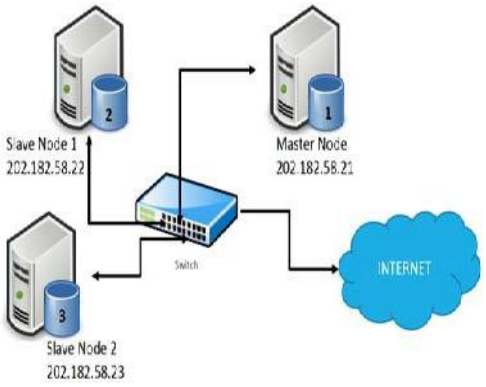


Gambar 2. Desain system

B. Desain Big Data Server

Dalam membangun platform Big Data digunakan tiga buah komputer server untuk membangun server Big Data dengan skema multi node cluster. Master node akan berperan sebagai titik sentral, yang mengatur proses pendistribusian data, pengalokasian cpu dan memory untuk job-job yang akan dijalankan. Dengan implementasi multi node cluster permasalahan akan terbatasnya ruang penyimpanan, cpu, maupun memory dapat teratasi. Nantinya jika ketiga server masih kurang, dapat ditingkatkan hanya dengan menambahkan node pada sisi slave sehingga tidak akan mengganggu sistem secara keseluruhan. Dengan impelementasi multi node cluster ini, kemungkinan akan adanya

data yang rusak dapat teratasi karena sifat dari Hadoop yang melakukan replikasi data pada semua node. Sementara Hadoop single-node diimplementasikan pada satu mesin. Mesin tersebut didesain menjadi master tetapi dapat bekerja juga sebagai slave dan semua proses distribusi dilakukan dalam satu mesin tersebut. Pada Hadoop terbagi menjadi dua layer yaitu layer HDFS yang menjalankan Namenode dan Datanode sedangkan layer MapReduce yang menjalankan Jobtracker dan Tasktracker. Kedua layer ini sangat penting terutama Namenode dan Jobtracker, karena apabila dua bagian ini tidak berjalan maka kerja HDFS dan Mapreduce tidak bisa dijalankan. Pada mesin single node, Datanode dan Tasktracker hanya ada satu, jika memiliki mesin yang banyak maka kedua bagian ini terbentuk pada setiap mesin (multinode). Gambar 3 menunjukkan desain infrastruktur server dengan skema multinode cluster.



Gambar 3. Desain infrastruktur server

Ketiga server menggunakan IP Public sehingga dapat diakses dari manapun, konfigurasi alamat IP pada ketiga server dapat dilihat pada Tabel 1. Throughput masing-masing node ke node lainnya adalah sekitar 1 Gbps.

Tabel 1. Konfigurasi alamat IP

Komputer Server	IP Address	Netmask	Gateway
Master Node	202.182.58.21	255.255.255.0	202.182.58.1
Slave Node 1	202.182.58.22	255.255.255.0	202.182.58.1
Slave Node 2	202.182.58.23	255.255.255.0	202.182.58.1

III. Hasil dan Pembahasan

Beberapa eksperimen yang dilakukan adalah uji coba program yang telah dibangun pada tiap lapisan dari Internet of Things dan penentuan performa dari algoritma klasifikasi yang digunakan pada Big Data Analytic yaitu Decision Tree dan Support Vector Machine.

A. Hasil uji coba program

Pada tahap ini akan dilakukan uji coba menjalankan program pada tiap lapisan dari platform Internet of Things yang telah dibuat. Program-program yang dijalankan mewakili dari tiap lapisan dari 7 lapisan platform IoT yang dibangun.

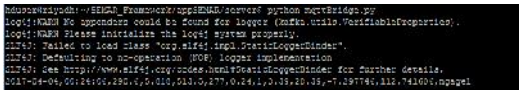
1. Lapisan *Physical Devices & Controllers*

Gambar 4 menunjukkan aplikasi yang berjalan pada node sensor. Data yang ditampilkan merupakan data gabungan yang akan dikirim ke server menggunakan protokol MQTT. Gambar 4 mewakili aplikasi yang dibangun pada lapisan ke-1 dari platform IoT yang dibangun.

Gambar 4. Aplikasi yang berjalan pada node sensor

2. Lapisan *Edge Computing*

Gambar 5 menunjukkan data yang diterima oleh MQTT Broker di sisi komputer server. Dari gambar 18 terlihat bahwa data masuk dari beberapa titik node sensor yang berbeda yaitu Ngagel, Pintu Air Jagir, Prapatan Prapen, Stikom, Joyoboyo, Wonorejo dan Karang Pilang. Gambar 6 mewakili aplikasi yang berjalan pada lapisan ke-3 dari platform IoT yang dibangun.



Gambar 5. Aplikasi pada MQTT Broker.

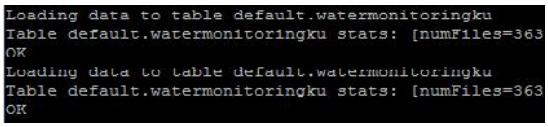
3. Lapisan *Data Accumulation*

Gambar 6 menunjukkan proses penyimpanan data pada Hadoop HDFS menggunakan *query* Hive. *Query* Hive dieksekusi menggunakan Spark SQL. Info yang ditampilkan menunjukkan data yang

diterima dari Kafka *Broker* dan kemudian disimpan pada HDFS. Gambar 6 mewakili aplikasi yang berjalan pada lapisan ke-4 dari platform IoT yang dibangun.

Gambar 6. Aplikasi penyimpanan data pada Hadoop HDFS

Gambar 7 merupakan informasi tambahan pada proses *background* tentang keberhasilan *load* data ke dalam Hadoop HDFS.



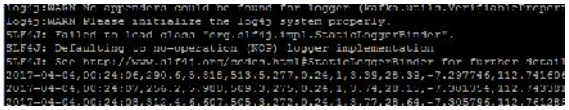
Gambar 7. Notifikasi *load* data ke tabel HDFS

Gambar 8 merupakan proses query data secara manual menggunakan *Command Line Interface* Hive untuk melihat data yang tersimpan pada Hadoop HDFS. Proses ini dilakukan secara *batch*. Untuk memasuki mode Hive CLI cukup dengan mengetikkan command ‘hive’ pada terminal linux.

Gambar 8. Data yang tersimpan pada Hadoop HDFS

4. Lapisan *Data Abstraction*

Gambar 9 merupakan proses aliran data menuju ke proses klasifikasi realtime. Pada gambar 22 terlihat aliran data yang diterima dari node yang akan didistribusikan ke klasifikasi realtime. Gambar 9 mewakili aplikasi yang dibangun pada lapisan ke-5 dari platform IoT yang dibangun.



Gambar 9. Aliran data menuju klasifikasi realtime

5. Lapisan *Application*

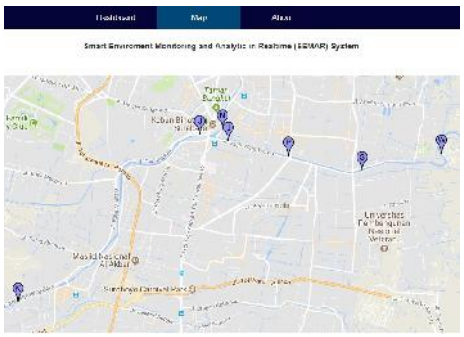
Gambar 10 merupakan hasil dari klasifikasi realtime. Info yang ditampilkan adalah data air yang telah diklasifikasi yang kemudian dikirim ke Kafka Broker menggunakan Kafka Producer. Gambar ini mewakili proses klasifikasi realtime yang terdapat pada lapisan ke-6 dari platform IoT yang dibangun.

Gambar 10. Proses pengiriman hasil klasifikasi data air ke Kafka Broker.

Gambar 11 merupakan data yang diterima oleh Node JS yang akan diproses pada front end untuk divisualisasi. Gambar 11 mewakili proses visualisasi data yang terdapat pada lapisan ke-6 (Application) dari platform IoT yang dibangun.

Gambar 11. Data yang diterima oleh backend Node JS

Gambar 12 menunjukkan hasil visualisasi pada front end. Port yang digunakan adalah port standar http yaitu 80. Pada gambar 12 terlihat posisi node sensor yang berada di beberapa titik lokasi sepanjang sungai di Kota Surabaya.



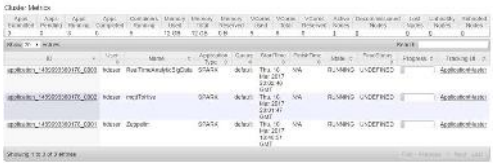
Gambar 12. Visualisasi lokasi node sensor

Gambar 13 menunjukkan chart dari data sensor dan hasil klasifikasi realtime dari sistem.

Gambar 13. Visualisasi data sensor dan hasil klasifikasi

Gambar 14 menunjukkan informasi node dan aplikasi yang berjalan di atas Hadoop dengan

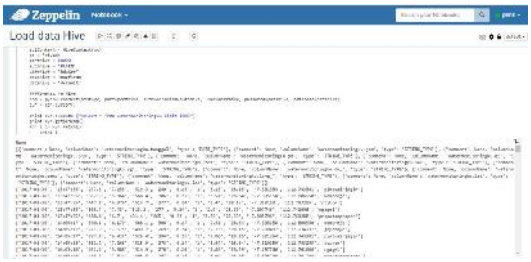
web user interface. Halaman ini dapat diakses pada port 8088.



Gambar 14. Informasi *job* pada Hadoop *cluster*

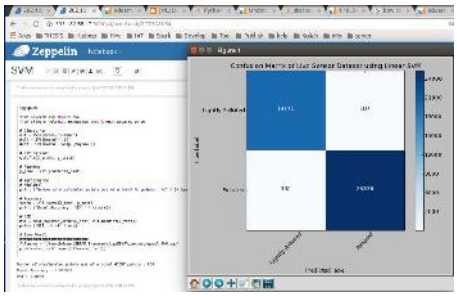
6. Lapisan *Collaboration & Processes*

Gambar 1 5 menunjukkan proses interaksi antara analis dan sistem. Informasi yang ditunjukkan berupa data air yang tersimpan pada Hadoop HDFS yang dapat diakses untuk analisa lebih lanjut. Pengaksesan data yang tersimpan pada Hadoop HDFS dapat dilakukan secara langsung. Zeppelin pada penelitian ini diakses melalui port 3000. Gambar 15 ini mewakili proses pada lapisan ke-7 dari platform IoT yang dibangun.



Gambar 14. Akses data pada Hadoop HDFS menggunakan *web interface*

Gambar 16 menunjukkan proses training dataset secara online dan interaktif. Proses training data pada proses ini juga dapat menghasilkan visualisasi dari proses *training*. Pada gambar 16 menunjukkan kode program python yang dijalankan untuk melakukan proses *training* dataset. Gambar 16 ini sekaligus mewakili aplikasi yang dibangun untuk proses pada layer ke-6 (lapisan Application), sub bagian proses *learning*.



Gambar 16. Proses *training* dataset

B. Hasil eksperimen pada algoritma klasifikasi

Hasil eksperimen tersebut dijabarkan dalam beberapa bagian yaitu: 1) *Confusion Matrix*, yang menampilkan data hasil *learning* yang berada pada kategori *True Positive* (TP), *False Positive* (FP), *True Negative* (TN) dan *False Negative* (FN); 2) Menentukan *Mean Squared Error* (MSE), *Mislabel*, dan *Score* dari data-data yang diperoleh dari *Confusion matrix*; 3) Grafik ROC yang menampilkan grafik perbandingan antara *True Positive Rate* dan *False Positive Rate*, yang bertujuan memvalidasi model yang telah dibangun; dan 4) *Processing time*, menghitung waktu rata-rata yang dibutuhkan server dalam mengolah, menganalisa, dan menampilkan data sensor air yang diterima dari node.

1. *Confusion Matrix* padaTabel 2 menunjukkan *confusion matrix* dari dataset uji laboratorium menggunakan Linear Support Vector Machine.

Tabel 2. Confusion Matrix dataset uji laboratorium menggunakan Linear SVM

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	206	6	0
	2	0	19	159	0
	3	0	0	0	0

Tabel 3 menunjukkan confusion matrix dari dataset uji laboratorium menggunakan Decision Tree.

Tabel 3. Confusion matrix dataset uji laboratorium menggunakan Decision Tree

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	212	0	0
	2	0	2	176	0
	3	0	0	0	0

Tabel 4 menunjukkan confusion matrix dari dataset sensor live menggunakan Linear Support Vector Machine.

Tabel 4. Confusion matrix dataset sensor live menggunakan linear SVM

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	19770	107	0
	2	0	182	25338	0
	3	0	0	0	0

Tabel 5 menunjukkan confusion matrix dari dataset sensor live menggunakan Decision Tree.

Tabel 5 *Confusion matrix* dataset sensor *live* menggunakan Decision Tree

		Predicted Class			
		0	1	2	3
Actual Class	0	0	0	0	0
	1	0	19859	18	0
	2	0	16	25504	0
	3	0	0	0	0

2. MSE, Mislabel, dan Score

Dalam melakukan klasifikasi, sebuah sistem diharapkan dapat melakukan klasifikasi semua set data dengan benar. Sebenarnya, kinerja suatu sistem klasifikasi tidak bisa bekerja secara 100% benar, namun kinerja sistem klasifikasi dapat diukur. Dalam mengukur kinerja sistem klasifikasi dapat menggunakan confusion matrix.

Dari confusion matrix, kita dapat menghitung mislabel, score dan mean squared error (MSE) pada tiap algoritma pada tiap dataset. Mislabel adalah hasil jumlah data yang diprediksi secara salah terhadap jumlah dataset yang akan diprediksi setelah dilakukan training data. Score adalah nilai akurasi hasil prediksi untuk mengetahui jumlah data yang diklasifikasikan secara benar. Untuk menghitung akurasi digunakan persamaan 1.

$$Score = \frac{Jumlah\ data\ yang\ diprediksi\ benar}{Jumlah\ data\ prediksi\ total}$$

(1)

MSE adalah nilai *error rate* yang digunakan untuk mengetahui jumlah data yang diklasifikasikan secara salah sehingga mengetahui laju *error* pada prediksi yang dilakukan. Untuk menghitung nilai *error rate* digunakan persamaan 2.

$$MSE = \frac{Jumlah\ data\ yang\ diprediksi\ salah}{Jumlah\ data\ prediksi\ total}$$

(2)

Berdasarkan hasil eksperimen yang dilakukan terhadap semua dataset dengan perbandingan data training dan data testing sebesar 70%:30%, diperoleh tabel nilai mislabel, score dan MSE yang dapat dilihat pada Tabel 6

Tabel 6. Nilai *mislabel, score* dan MSE

Features	Dataset	Algorithm	Mislabel	Score	MSE
pH, TSS, DO, Temp, Turbidity	Laboratory test	Linear Support Vector	24 / 390	0.938462	0.0615
		Decision Tree	2 / 390	0.994872	0.0051
	Live Sensor	Linear Support Vector	286 / 45397	0.993700	0.0063
		Decision Tree	34 / 45397	0.999251	0.0007

Pada Tabel 7 terlihat jika secara umum performa algoritma Decision Tree lebih baik dibandingkan dengan algoritma Support Vector Machine pada penelitian ini. Algoritma Decision Tree memiliki rata-rata tingkat akurasi sebesar 0.9970615 (99.7%) dan MSE rata-rata 0.0029. Support Vector Machine memiliki rata-rata tingkat akurasi sebesar 0.966081 (96.6%) dan MSE rata-rata 0.0339.

3. ROC

Kurva ROC merupakan salah satu cara melakukan analisa terhadap model klasifikasi yang telah dibuat dalam menentukan parameter model yang diinginkan sesuai dengan karakteristik dari model klasifikasi yang diinginkan. Penggunaan kurva ROC seringkali digunakan dalam mengevaluasi proses klasifikasi, dikarenakan mempunyai kemampuan evaluasi secara menyeluruh dan cukup baik. Misalkan pada table yang terdiri dari dua buah kelas data yaitu data kelas yang dihasilkan dari classifier (Predicted Class) dan data kelas asli yang telah diketahui (Actual Class) dimana data Predicted Class yang sama dengan Actual Class maka termasuk True Positive (TP), sedangkan data Predicted Class yang tidak sama dengan Actual Class tapi termasuk dari data hasil klasifikasi maka termasuk False Positive (FP).

Kurva ROC digunakan sebagai grafik perbandingan antara True Positive Rate (TPR) pada sumbu vertikal dengan False Positive Rate (FPR) pada sumbu horisontal. TPR merupakan proporsi data yang digunakan positif yang teridentifikasi dengan benar antara data Predicted Class dengan Actual Class sedangkan FPR merupakan proporsi data negatif yang teridentifikasi salah sebagai positif pada suatu model klasifikasi. True Positive Rate dan False Positive Rate, dapat dihitung menggunakan persamaan:

$$TPR = \frac{TP}{TP + FP}$$

(3)

$$FPR = \frac{FP}{TP+FP}$$

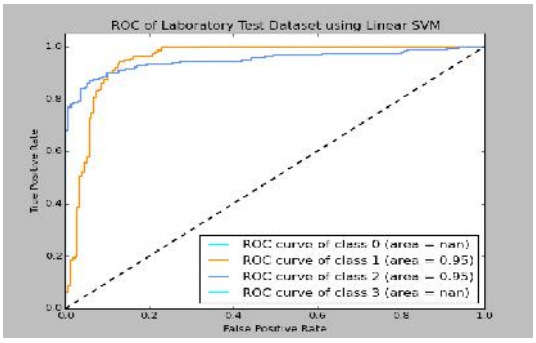
(4)

Pada kurva ROC, luas area dibawah kurva dikenal dengan AUC (Area Under the ROC Curve), dimana nilai AUC berkisaran antara 0 sampai dengan 1, semakin mendekati 1 maka semakin baik nilai uji pada karakteristik klasifikasi tersebut. Nilai kategori AUC pada Tabel 7.

Tabel 7 Tabel nilai AUC

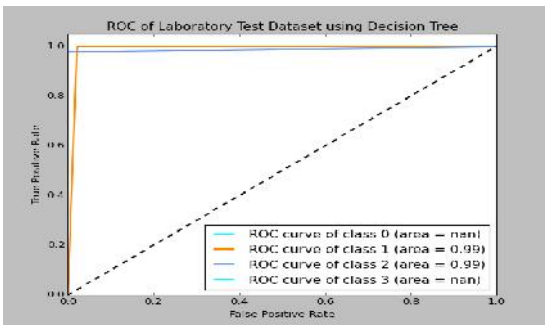
Range Nilai AUC	Keterangan
0.5 – 0.6	Fail
0.6 – 0.7	Poor
0.7 – 0.8	Fair
0.8 – 0.9	Good
0.9 – 1.0	Excellent

Gambar 11 menunjukkan grafik ROC dari dataset uji laboratorium menggunakan Linear SVM



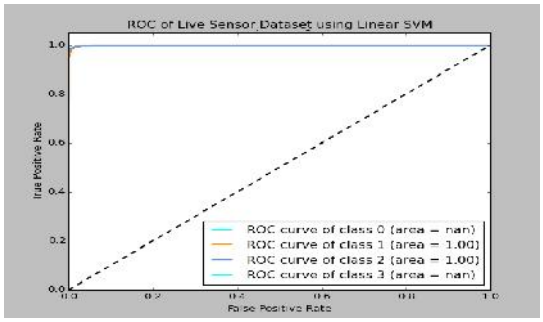
Gambar 11 ROC dataset uji laboratorium menggunakan Linear SVM

Gambar 12 menunjukkan grafik ROC dari dataset uji laboratorium menggunakan Decision Tree.



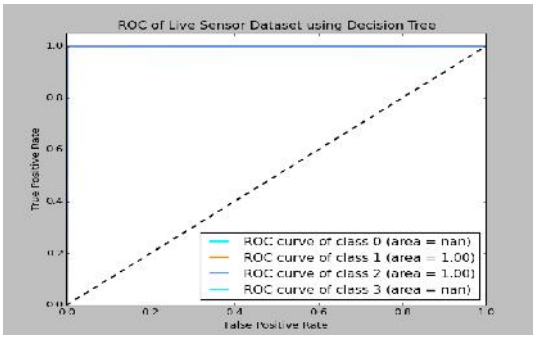
Gambar 12. ROC dataset uji laboratorium menggunakan Decision Tree

Gambar 13 menunjukkan grafik ROC pada dataset sensor live menggunakan Linear SVM.



Gambar 13. ROC dataset sensor live menggunakan linear SVM

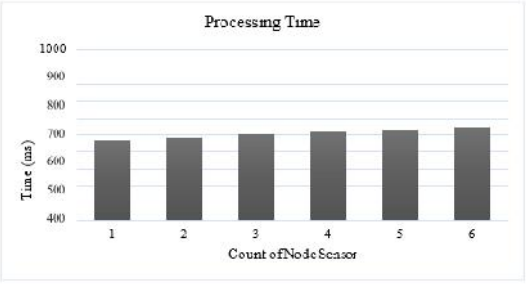
Gambar 14 menunjukkan grafik ROC pada dataset sensor live menggunakan Decision Tree.



Gambar 14. ROC dataset sensor *live* menggunakan Decision Tree

4. Hasil Uji Processing Time

Dalam menentukan processing time dihitung sejak data tersebut diterima oleh server hingga data tersebut divisualisasi pada front end. Selain itu dalam perhitungan processing time ini digunakan skenario jumlah node yang mengirim data secara bersamaan yang berbeda-beda. Gambar 15 menunjukkan rata-rata waktu pemrosesan data pada server.



Gambar 15. *Processing time* dalam mS vs jumlah node sensor

C. Analisis Hasil Eksperimen

Dari *confusion matrix* yang dihasilkan pada percobaan algoritma klasifikasi diperoleh hasil yang memperlihatkan tidak adanya sampel data yang diperoleh baik dari uji laboratorium PDAM maupun dari sensor live PDAM yang memenuhi standar mutu baik air minum. Di samping juga tidak ditemukan sampel data yang memenuhi standar tersemar berat. Hal ini menunjukkan bahwa sungai- sungai di sepanjang sungai di Kota Surabaya berada pada tingkat tercemar ringan dan tercemar sedang. Hal ini merupakan informasi awal yang dapat dijadikan acuan untuk meningkatkan kualitas air sungai di Kota Surabaya atau malah membiarkan hingga air sungai di Kota Surabaya berada pada level tercemar berat. Hasil learning antara algoritma Linear SVM dan Decision Tree menunjukkan performa yang baik dimana tingkat akurasi berada pada level > 90% dan MSE sekitar 0.019075. Hal ini menunjukkan dataset yang digunakan tidak mengalami fluktuasi angka yang signifikan. Namun jika kedua algoritma dibandingkan maka algoritma Decision Tree menghasilkan akurasi yang lebih baik yaitu dengan score 0.999251 untuk dataset sensor *live* dan 0.994872 untuk dataset uji laboratorium. Salah satu penyebab performa Linear SVM yang kurang baik dibanding dengan Decision Tree adalah rapatnya jarak antar data sensor yang ada, hal ini menyebabkan algoritma Linear SVM kesulitan dalam menemukan garis *hyperplane* yang paling optimal.

Dari hasil grafik ROC menunjukkan bahwa kinerja sistem klasifikasi yang dibangun memiliki performa yang ‘excellent’. Dimana status ‘excellent’ tersebut diperoleh dari nilai grafik ROC yang diperoleh > 0.9 dan bahkan mendekati atau sama dengan 1.

Spark merupakan aplikasi machine learning pada Big Data yang menggunakan skema pemrosesan dalam memori. Hal ini mengakibatkan kebutuhan penggunaan memori yang besar oleh Spark. Dalam menjalankan aplikasi Spark pada Big Data platform, terkadang ditemukan kendala *error* yaitu *Out of Memory Error*. Hal ini dapat diatasi dengan meningkatkan alokasi Java Heap pada sistem. Dalam percobaan ini, kami menggunakan Yarn untuk memanajemen cpu dan memori pada *job* Spark. Pada yarn, kami alokasikan 4 GB memori yang dialokasikan pada tiap *job* Spark yang berjalan.

Server rata-rata hanya membutuhkan waktu 508 miliseconds untuk memproses data yang berasal dari semua node. Waktu pemrosesan tersebut adalah waktu di saat data diterima oleh server dan berakhir saat data telah divisualisasi. Hal ini bisa terjadi karena skema *direct flow* yang digunakan dalam sistem ini. Hal ini didukung oleh kemampuan Spark dalam memproses data dengan skema pemrosesan dalam memori dan kemampuan Node JS yang dapat mendukung pemrosesan data *realtime* hingga divisualisasi. *Processing time* di sini tidak termasuk waktu transmisi data dari node sensor ke server. Dari hasil percobaan, diperoleh hasil jika rata-rata waktu pengiriman data dari node ke server adalah 1 detik. Hal ini menunjukkan bahwa sistem yang dibangun efektif.

IV. Kesimpulan

Pada penelitian ini bertujuan untuk membangun sistem klasifikasi *realtime* air sungai dengan menggunakan teknologi IoT dan *Big Data*. Dimana dalam pengeimplementasiannya dikembangkan sebuah platform IoT yang terdiri dari 7 lapisan IoT dan diintegrasikan dengan teknologi *Big Data* pada beberapa lapisan tersebut. Penggunaan teknologi *Big Data* utamanya berada pada sisi penyimpanan, analisa dengan klasifikasi, dan manajemen data dalam server.

Hasil klasifikasi dengan menggunakan algoritma Linear SVM dan Decision Tree menunjukkan performa yang baik, dimana akurasinya berada pada level di atas 90%. Pada dataset uji laboratorium algoritma Linear SVM menunjukkan akurasi sebesar 0.935897 sedangkan Decision Tree 0.994872. Sementara untuk dataset sensior live algoritma Linear SVM menunjukkan akurasi sebesar 0.993634 sedangkan Decision Tree 0.999251. Hal ini dapat disimpulkan bahwa algoritma Decision Tree memiliki akurasi yang lebih baik dibandingkan dengan algoritma Linear SVM. Dimana algoritma Decision Tree memiliki rata-rata tingkat akurasi sebesar 0.9970615 dan algoritma Linear SVM sebesar 0.9647655.

Pengujian hasil validasi yang telah dilakukan pada dataset uji laboratorium maupun dataset sensor live berdasarkan grafik ROC dengan nilai Area Under ROC menunjukkan di atas angka 0.9. Dengan demikian dapat dikatakan bahwa unjuk kerja nilai Area Under ROC menunjukkan kinerja 'Excellent'.

Sistem yang dibangun hanya membutuhkan rata-rata 508 miliseconds dalam memproses data oleh server yang diterima dari node sensor. Hal ini menunjukkan sistem platform IoT-Big Data yang dibangun memiliki kinerja yang sangat baik.

Ucapan Terima Kasih

Ucapan terima kasih diberikan kepada Kementerian Riset Teknologi dan Pendidikan Tinggi yang telah memberikan beasiswa pascasarjana di Politeknik Elektronika Negeri Surabaya.

Daftar Pustaka

[1] *Sukaridhoto, Sritrusta, Dadet Pramadihanto, Muhammad Alif, Andrie Yuwono, and Nobuo Funabiki. A design of radio-controlled submarine modification for river water quality monitoring, In Intelligent Technology and Its Applications (ISITIA), 2015 International Seminar on, pp. 75-80. IEEE, 2015.*

[2] *Yuliandoko, Herman, Sritrusta Sukaridhoto, M. Udin Harun Al Rasyid, and Nobuo Funabiki. Performance of Implementation IBR-DTN and Batman- Adv Routing Protocol in Wireless Mesh Networks,EMITTER International Journal of Engineering Technology 3, no. 1 (2016).*

[3] *Sukaridhoto, Sritrusta, Rahardhita Widyatra Sudibyo, Widi Sarinastiti, Rizky Dharmawan, Atit Sasono, Ahmad Andika Saputra, and Shiori Sasaki. Design and development of a portable low-cost COTS-based water quality monitoring system, In Intelligent Technology and Its Applications (ISITIA), 2016 International Seminar on, pp. 635-640. IEEE, 2016.*

[4] *Abdillah, Abid, Muhammad Herwindra, Yohanes Panduman, Muhammad Akbar, Marlanisa Afifah, Sritrusta Sukaridhoto, Shiori Sasaki. Design and Development Low Cost Coral Monitoring System for Shallow Water Based on Internet of Underwater Things, In Advanced Reseacrh in Electronic Engineering and Information Technology International Conference (AVAREIT), 2016.*

[5] *Berlian, Muhammad Herwindra, Tegar Esa Rindang Sahputra, Buyung Jofi Wahana Ardi, Luhung Wahya Dzatmika, Adnan Rachmat Anom Besari, Rahardhita Widyatra Sudibyo, and Sritrusta Sukaridhoto. Design and implementation of smart environment monitoring and analytics in real- time system framework based on internet of underwater things and big data." In Electronics Symposium (IES), 2016 International, pp. 403-408. IEEE, 2016.*

[6] *Modaresi, Fereshteh, and Shahab Araghinejad. A comparative assessment of support vector machines, probabilistic neural networks, and K-nearest neighbor algorithms for water quality classification, Water resources management 28, no. 12 (2014): 4095-4111.*

[7] *Ladjal, Mohamed, Mohamed Bouamar, Mohamed Djerioui, and Youcef Brik. Performance evaluation of ANN and SVM multiclass models for intelligent water quality classification using Dempster-Shafer Theory, In Electrical and Information Technologies (ICEIT), 2016 International Conference on, pp. 191-196. IEEE, 2016.*

[8] *Jaloree, Shailesh, Anil Rajput, and Sanjeev Gour. Decision tree approach to build a model for water quality, Binary Journal of Data Mining & Networking 4, no. 1 (2014): 25-28*

[9] *Saghebian, S. Mehdi, M. Taghi Sattari, Rasoul Mirabbasi, and Mahesh Pal. Ground water quality classification by decision tree method in Ardebil region, Iran, Arabian Journal of Geosciences 7, no. 11 (2014): 4767-4777.*

[10] *Fazio, Maria, Antonio Celesti, Antonio Puliafito, and Massimo Villari. Big data storage in the cloud for smart environment monitoring, Procedia Computer Science 52 (2015): 500-506.*

[11] *Meng, Xiangrui, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman et al., Mllib: Machine learning in apache spark, Journal of Machine Learning Research 17, no. 34 (2016): 1-7.*

[12] *Richter, Aaron N., Taghi M. Khoshgoftaar, Sara Landset, and Tawfiq Hasanin., A multi-dimensional comparison of toolkits for machine learning with big data, In Information Reuse and Integration (IRI), 2015 IEEE International Conference on, pp. 1-8. IEEE, 2015.*

[13] *Kalaria, Dhruv, MQTTKafkaBridge: Bridge MQTT Messages from Mosquitto to Kafka Broker on the same topic, http://github.com/DhruvKalaria/MQTTKafkaBridge, diakses 28 April 2017.*