

# Klasifikasi Jenis Pemeliharaan dan Perawatan *Container Crane* menggunakan Algoritma *Machine Learning*

Via Ardianto Nugroho, Derry Pramono Adi, Achmad Teguh Wibowo, MY Teguh Sulistyono, and Agustinus Bimo Gumelar

**Abstract**— In the container service industry, Nilam Terminal is a customer of PT. BIMA, which is specifically engaged in repair and maintenance services for heavy equipment. This terminal is the central place to carry out loading and unloading domestic container activities, four container cranes to serve two ships. The maintenance process for heavy equipment such as container cranes, which have been operating so far, seems to have paid less attention to data on the grouping or classification of the types of maintenance required by the heavy equipment. In the future, heavy equipment may show underperformance that can even lead to work accidents. Besides, negligence in maintaining container crane can also cause further maintenance costs. The production target for loading and unloading can be reduced, and delays in docking ships' schedules are also possible. Machine learning methods or commonly known as Machine Learning (ML), can quickly eliminate these possibilities. In this study, we designed ML to identify and then group the appropriate type of container crane maintenance, namely light or heavy. The ML method chosen for use in this study is Random Forest, Support Vector Machine, k-Nearest Neighbor, Naïve Bayes, Logistic Regression, J48, and Decision Tree. This study shows the ML model tree's success in learning the type of container crane maintenance data (numerical and categorical), with J48 showing the best performance with accuracy and ROC-AUC values reaching 99.1%. We carry out classification considerations by referring to the last maintenance date, hour meter, breakdown, shutdown, and spare parts.

**Index Terms**— Classification; Container Crane Maintenance; J48; K-Nearest Neighbor; Machine Learning;

**Abstrak**— Pada industri jasa pelayanan peti kemas, Terminal Nilam merupakan pelanggan dari PT. BIMA, yang secara khusus bergerak di bidang jasa perbaikan dan perawatan alat berat. Terminal ini menjadi sentral tempat untuk melakukan aktifitas bongkar muat peti kemas domestik yang memiliki empat buah *container crane* untuk melayani dua kapal. Proses perawatan alat berat seperti *container crane* yang selama ini beroperasi, agaknya kurang memperhatikan data pengelompokan atau klasifikasi jenis perawatan yang dibutuhkan oleh alat berat tersebut. Di kemudian hari, alat berat dapat menunjukkan kinerja yang tidak maksimal bahkan dapat berujung pada kecelakaan kerja. Selain itu, kelalaian perawatan *container crane* juga dapat menyebabkan pembengkakan biaya perawatan lanjut. Target produksi bongkar muat dapat berkurang dan juga keterlambatan jadwal kapal sandar sangat mungkin terjadi. Metode pembelajaran *Machine Learning* (ML), dengan mudah dapat melenyapkan kemungkinan-kemungkinan tersebut. ML dalam penelitian ini, kami rancang agar bekerja dengan mengidentifikasi lalu mengelompokkan jenis perawatan *container crane* yang sesuai, yaitu ringan atau berat. Metode ML yang pilih untuk digunakan dalam penelitian ini yaitu *Random Forest*, *Support Vector Machine*, *k-Nearest Neighbor*, *Naïve Bayes*, *Logistic Regression*, *J48*, dan *Decision Tree*. Penelitian ini menunjukkan keberhasilan ML model *tree* dalam melakukan pembelajaran jenis data perawatan *container crane* (numerik dan kategoris), dengan J48 menunjukkan performa terbaik dengan nilai akurasi dan nilai ROC-AUC mencapai 99,1%. Pertimbangan klasifikasi kami lakukan dengan mengacu kepada tanggal terakhir perawatan, *hour meter*, *breakdown*, *shutdown*, dan *sparepart*.

**Kata Kunci**—J48; Klasifikasi; K-Nearest Neighbor; Machine Learning; Perawatan Container Crane;

Via Ardianto Nugroho is with the Sistem Informasi, Fakultas Ilmu Komputer, Universitas Narotama, Surabaya, Indonesia (email: via.ardi@gmail.com).

Derry Pramono Adi is with the Sistem Informasi, Fakultas Ilmu Komputer, Universitas Narotama, Surabaya, Indonesia (e-mail: derryalbertus@ieee.org).

Achmad Teguh Wibowo is with the Fakultas Sains dan Teknologi, UIN Sunan Ampel, Surabaya, Indonesia (e-mail: atw@uinsby.ac.id).

MY Teguh Sulistyono is with the Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang, Indonesia (e-mail: teguh.sulistyono@dsn.dinus.ac.id).

Agustinus Bimo Gumelar is with the Fakultas Ilmu Komputer, Universitas Narotama, Surabaya, Indonesia (corresponding author, email: bimogumelar@ieee.org)

## I. PENDAHULUAN

Terminal Nilam memiliki empat buah alat berat *container crane*, yang mana dapat melayani dua kapal dengan masing-masing dua *container crane*. Apabila alat tersebut mengalami kerusakan, maka akan berdampak pada kinerja di lapangan. Kapal yang seharusnya di layani bongkar muat dua *crane*, terpaksa dilayani hanya satu *crane*. Secara tidak langsung, kejadian ini akan mempengaruhi jadwal sandar kapal selanjutnya dan target produksi bongkar muat [1]–[3].

Selain produktifitas bongkar muat kapal, proses perawatan juga berdampak pada antrian kapal sandar. Proses perawatan yang tidak tepat mengakibatkan pada jadwal dan waktu tunggu kapal sandar yang sudah terjadwal dapat berubah tidak sesuai dengan sistem penjadwalan awal yang dibuat oleh manajemen. Sehingga manajemen harus membuat jadwal ulang untuk kapal yang akan bersandar di dermaga pelabuhan serta waktu tunggu kapal sandar menjadi lebih lama. Kejadian tersebut patut menyebabkan pembengkakan biaya perawatan *container crane* [4]–[6].

Persiapan perancangan keseluruhan sistem informasi pengambilan keputusan atau sistem pakar untuk mengatur jadwal perawatan *container crane* serta jenis perawatan yang diperlukan hendaknya memenuhi syarat untuk mengidentifikasi jenis perawatan. *Profiling* fisik *container crane* serta variabel-variabel catatan penggunaan *container crane* dapat ditinjau sebagai data. Maka, *Machine Learning* (ML) yang merupakan metode *data-driven*, sangatlah cocok untuk untuk *task* klasifikasi jenis perawatan yang diperlukan sebuah *container crane* [1], [7].

Sebagai metode matematis-statistik, ML memberi insight baik dalam kinerja klasifikasi. Cannas dkk menggunakan *Naïve Bayes* (NB) untuk melakukan klasifikasi serta prediksi *delay* jadwal kapal sandar, dengan total akurasi mencapai 67,3% [8]. Viellechner dan Spinler juga memiliki *concern* terhadap *delay* jadwal kapal sandar dengan kaitannya *supply chain* [9]. Dalam penelitian mereka, Viellechner menggunakan *Logistic Regression* (LogR) sebagai salah satu metode pembelajaran klasifikasi keterlambatan kapal sandar, yang mana mencapai angka akurasi 75% [9]. Maka, pendekatan matematis-statistik sangat dapat diaplikasikan pada data berjenis numerik.

Selain pendekatan matematis-statistik, ML juga memiliki modifikasi model, yaitu *tree-based model*. Model ini dimulai dari *Decision Tree* (DT), yang mana selanjutnya dikembangkan menjadi beberapa metode. DT adalah salah satu pendekatan pemodelan prediktif yang digunakan dalam statistik, *data mining*, dan ML. Secara umum, DT sebagai model prediktif mengamati suatu *item* (diwakili sebagai “cabang”) lalu menilai *item* tersebut (diwakili sebagai “daun” dan seusai dengan target yang ditentukan). DT memodelkan dimana variabel target dapat mengambil sekumpulan nilai diskrit yang disebut sebagai *classification trees* [10].

Cannas dkk juga menggunakan *Random Forest* (RF), suatu *tree-based* ML yang cukup terkenal, dengan nilai akurasi yang lebih besar yaitu 70,3% [8]. Dovbysh dkk meneliti kegunaan *tree-based model* dalam data kategoris, yang mana memunculkan angka akurasi 91,6% untuk metode J48 [11]. Dengan tujuan kualitatif seperti pada penelitian ini, Zareiforush dkk melakukan klasifikasi grade butir nasi menggunakan metode J48, yang mana memunculkan angka 0.1184 untuk *Root Mean Square Error* (RMSE) [12]. Maka, ML *tree-based model* dapat diaplikasikan dengan data kategoris.

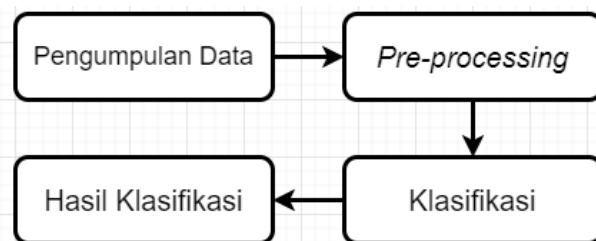
Melalui data resmi dan metode ML yang ada, serta menggunakan kombinasi data numerik dan kategoris,

kami meneliti performa ML genre matematis-statistik dan *tree-based model*, dalam melakukan klasifikasi jenis perawatan terbaik bagi *container crane*. Adapun kategori perawatan meliputi pilihan biner: ringan dan berat. Sepengetahuan kami, penelitian ini bersifat preliminary terhadap perawatan *container crane*, namun keterbatasan data memotivasi jaranganya pernyataan kualitatif dan eksperimen klasifikasi antara perawatan ringan atau berat.

Makalah ini disusun sebagai berikut: Bagian 1 menjelaskan pendahuluan, yaitu kepentingan perawatan *container crane* menggunakan metode ML. Bagian 2 menjelaskan beberapa penelitian klasifikasi lainnya yang sukses menggunakan metode ML seperti pada penelitian ini. Bagian 3 menjabarkan alur penelitian yang kami lakukan, serta hasil eksperimen dalam bentuk perbandingan antar metode ML. Bagian 4 menarik kesimpulan dari keseluruhan eksperimen.

## II. METODE PENELITIAN

Pada bagian ini, dijelaskan rincian metode klasifikasi ML yang kami gunakan, serta alur penelitian yang mana menggunakan metode tersebut. Gambar 1 menunjukkan alur penelitian dari klasifikasi jenis perawatan *container crane*. Mengurai Gambar 1, kami melakukan pengumpulan data dan memilih variabel/atribut yang memiliki data lengkap. Pengumpulan data dilakukan dengan mengambil data primer dari kegiatan operasional PT. BIMA. Kemudian, kami melakukan *pre-processing* dengan menghilangkan *noise* menggunakan package *Replace Missing Values* [13]. Setiap metode klasifikasi kemudian dijalankan pada program WEKA [14], dengan rata-rata proses klasifikasi mengeluarkan hasil waktu proses kurang dari satu detik.



Gambar. 1. Alur Penelitian Klasifikasi Perawatan *Container Crane*

### A. Material dan Pengumpulan Data

Data yang digunakan adalah data primer yang diperoleh dari perusahaan PT. BIMA yang beroperasi di Terminal Nilam Multipurpose Pelabuhan Tanjung Perak Surabaya. Data memiliki periode selama satu tahun, dimulai dari Januari 2019 sampai dengan Desember 2019. Data yang didapat berupa data mentah *digital* yang dianalisa untuk menentukan variabel yang selanjutnya akan dijadikan obyek penelitian sebagaimana disajikan pada Tabel 1.

Table 1. Data Primer

Daftar Isi	Atribut Terpilih?
<i>Performance Chart</i>	Tidak
<i>Performance Report</i>	Tidak
<i>Daily Report</i>	Tidak
<i>Hour Meter</i>	Ya
<i>Fuel Consumption</i>	Tidak
<i>Spare Parts Used</i>	Ya
<i>Breakdown Report</i>	Ya
<i>Shutdown Report</i>	Ya
<i>Accident Report</i>	Tidak
<i>Maintenance Report</i>	Tidak

*Performance Chart* merupakan data grafik kinerja selama satu bulan, *performance report* adalah laporan data kinerja selama satu bulan, *daily report* adalah laporan harian setiap *shift* kerja, *hour meter* adalah data penggunaan *container crane* selama sehari dengan hitungan jam. Selibhnya, *fuel consumption* adalah data penggunaan bahan bakar perhari selama satu bulan, *spare part used* adalah data suku cadang yang digunakan, *breakdown* adalah data kerusakan, shutdown data kerusakan yang menyebabkan *crane* berhenti total sehingga tidak dapat beroperasi. Sementara *accident report* dan *maintenance report* adalah data kecelakaan pada *crane* dan laporan perawatan.

Beberapa data yang yang digunakan sesuai kebutuhan penelitian yaitu atribut tanggal terakhir perawatan, *hour meter* (HM), *breakdown* (BD), *shutdown* (SD), dan *sparepart* (SP). Atribut ini terpilih atas ketersediaan dan keterlengkapan data dari narasumber. Empat variabel terpilih seperti tertera pada Tabel 2.

Tabel 2. Atribut Terpilih

Atribut	Uraian	Tipe Data
HM	<i>Hour Meter</i>	<i>Real (Days)</i>
SD	<i>Shutdown</i>	<i>Real (Days)</i>
BD	<i>Breakdown</i>	<i>Real (Days)</i>
SP	<i>Spare Parts</i>	<i>Real (Days)</i>

Sebagai tambahan, Tabel 3 menunjukkan spesifikasi perangkat keras *personal computer* (PC) untuk eksperimen ini.

Tabel 3. Spesifikasi PC untuk Eksperimen

Hardware	Spesifikasi
CPU	Ryzen 7 3700X 3.6GHz 8-Core/16-Thread
GPU	Gigabyte GeForce GTX 1650 Super 4GB OC
RAM	Corsair Vengeance LPX 32GB DDR4
Storage	ADATA SX8200 Pro NVMeM.2 1TB

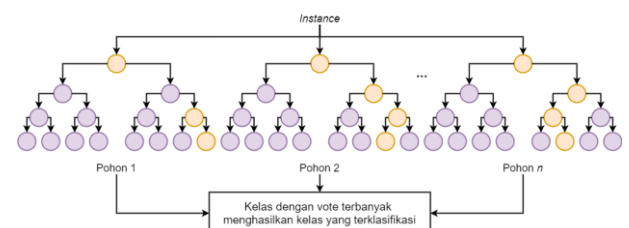
Terlebih, kami menggunakan WEKA sebagai tool ML utama kami [14]. Seluruh *package* ML yang dibutuhkan telah dipasang pada WEKA.

## B. Algoritma Pengklasifikasi

*Random Forest* (RF) bekerja dengan membangun *decision trees* lalu menggabungkannya. Jika digunakan dengan benar, algoritma RF bisa sangat berguna untuk tipe data regresi (numerik) maupun kategoris [15]. RF memiliki implementasi yang relatif mudah, waktu *training* yang cepat, dan dapat menemukan representasi akurat dari *decision trees* yang digunakan. Persamaan (1) menunjukkan kelas dan probabilitas untuk menentukan indeks Gini dari setiap cabang pada sebuah node, menentukan cabang mana yang lebih mungkin terjadi. Dalam rumus ini,  $p_i$  mewakili frekuensi relatif kelas dalam data dan  $c$  mewakili jumlah kelas [15].

$$Gini = 1 - \sum_i p_i^2 = 1 - \sum_i (p_i)^2 \quad (1)$$

Maka, RF cocok untuk digunakan pada data kelas biner berisi informasi numerik pada tiap atribut untuk jenis perawatan *container crane*. Gambar 2 menunjukkan ilustrasi sederhana dari RF [16].

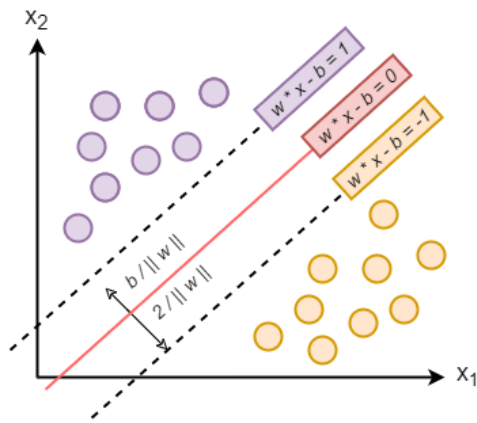
Gambar. 2. Ilustrasi *Random Forest* [16]

*Support Vector Machine* (SVM) dapat digunakan untuk tugas regresi dan klasifikasi. Namun, SVM banyak digunakan dalam tujuan klasifikasi. Tujuan dari SVM adalah untuk menentukan *hyperplane* dalam ruang dimensi- $N$  ( $N$  dikurangi jumlah fitur atau atribut) yang secara jelas mengklasifikasikan titik data atau *data points*. Persamaan (2) melibatkan penghitungan *inner products* dari vektor *input* baru ( $x$ ) dengan semua *support vector* dalam data *training*. Koefisien  $B_0$  dan  $a_i$  (untuk setiap *input*) harus diperkirakan dari data *training* dengan algoritma *learning*.

$$f(x) = B_0 + \text{sum}(a_i \times (x, x_i)) \quad (2)$$

Maka, SVM cocok untuk digunakan pada data kelas biner berisi informasi numerik pada tiap atribut untuk jenis perawatan *container crane*. Gambar 3 menunjukkan ilustrasi sederhana SVM [17].

*k-Nearest Neighbor* (k-NN) mengamati karakteristik dari data yang dikumpulkan dari data *training* dan *test* [18]. Secara umum, k-NN menghitung jarak antara titik data, menggunakan rumus Euclidean sederhana, seperti pada Persamaan (3). Sedangkan Gambar 4 menunjukkan ilustrasi sederhana k-NN [19].

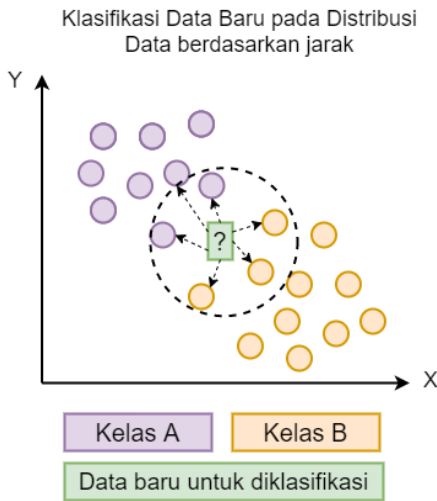


Gambar. 3. Ilustrasi Support Vector Machine [17]

Persamaan (3) menggunakan  $n$  sebagai sejumlah dimensi atau fitur dalam ML. Titik data yang terletak pada jarak minimum dari titik uji diasumsikan termasuk dalam kelas yang sama [20]. Maka, k-NN cocok untuk digunakan pada data kelas biner berisi informasi numerik pada tiap atribut untuk jenis perawatan *container crane*. Gambar 4 menunjukkan ilustrasi sederhana k-NN.

$$d(p, q) = d(q, p) = \sqrt{(q_1 - p_1)^2 + \dots + (q_n - p_n)^2} \quad (3)$$

$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$



Gambar. 4. Ilustrasi K-NN dari signifikansi dalam kalkulasi [19]

*Naïve Bayes* (NB) adalah sekumpulan algoritma klasifikasi berdasarkan *Bayes' Theorem*. NB bukan algoritma tunggal, melainkan sekumpulan algoritma yang semuanya memiliki prinsip yang sama, yaitu setiap pasangan fitur yang diklasifikasikan tidak bergantung satu sama lain [21], [22]. *Bayes' Theorem* mencari probabilitas suatu peristiwa terjadi dengan mempertimbangkan kemungkinan peristiwa lain yang telah terjadi. *Bayes' Theorem* dinyatakan secara matematis pada Persamaan (4).

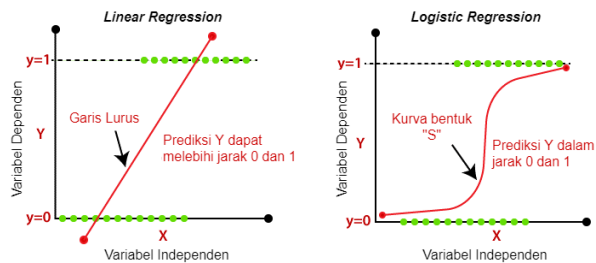
$$P(A|B) = P(B|A)P(A)/P(B) \quad (4)$$

Dimana A dan B adalah events, sedangkan  $P(B) \neq 0$ . Selanjutnya,  $P(A/B)$  dan  $P(B/A)$  adalah probabilitas kondisional, sedangkan  $P(A)$  dan  $P(B)$  adalah probabilitas marginal. NB berlaku untuk data diskrit. Dalam kasus data kontinu, diperlukan beberapa asumsi mengenai distribusi nilai dari setiap fitur. NB berbeda terutama karena asumsi terkait distribusi  $P(x_i|y)$ . Maka, NB cocok untuk digunakan pada data kelas biner berisi informasi numerik pada tiap atribut untuk jenis perawatan *container crane*.

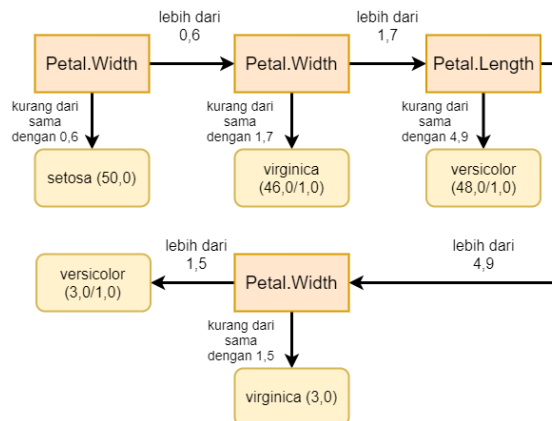
*Logistic Regression* (LogR), secara singkat, adalah algoritma ML untuk *task* klasifikasi. Terlebih, dengan diberikan satu *set* variabel independen, LogR bekerja sangat baik untuk klasifikasi kelas biner. LogR dapat dianggap sebagai kasus khusus dari regresi linier jika variabel hasil bersifat kategoris, di mana LogR justru menggunakan logaritma peluang sebagai variabel dependen [15], [23].

$$g(E(y)) = \alpha + \beta x^2 + \gamma x^2 \quad (5)$$

Persamaan (5) menunjukkan rumus umum dari LogR, dimana  $g$  adalah fungsi *link*,  $E(y)$  adalah ekspektasi dari target variabel,  $\alpha + \beta x^2 + \gamma x^2$  adalah *predictor* linear. Fungsi *link* adalah menghubungkan ekspektasi  $y$  dengan *predictor linear* [15], [24]. Maka, LogR cocok untuk digunakan pada data kelas biner berisi informasi numerik pada tiap atribut untuk jenis perawatan *container crane*. Gambar 5 menunjukkan ilustrasi sederhana LogR [25].



Gambar 5. Ilustrasi LogR [25]



Gambar. 6. Ilustrasi J48 pada IRIS [26]

Algoritma *decision tree* C4.5 milik Quinlan mengaktualisasikan J48 untuk membuat C4.5 versi “terpangkas” atau *trimmed*. Setiap aspek informasi dibagi menjadi beberapa subset kecil untuk dijadikan dasar keputusan. J48 melihat pada perolehan data standar yang benar-benar menghasilkan perpecahan informasi dengan memilih atribut [27].

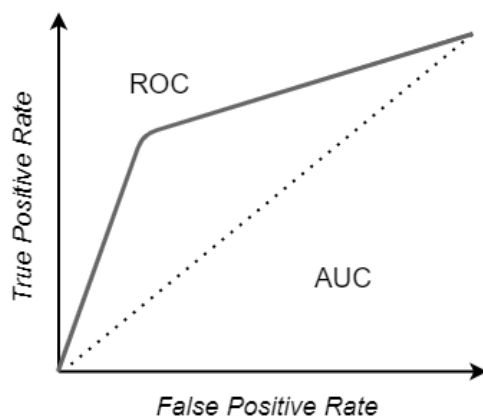
Tabel 4. Basis dari setiap Algoritma ML yang dipakai

Algoritma ML	Basis Algoritma
RF	<i>Tree-based Model</i>
SVM	<i>Tree-based Model</i>
k-NN	<i>Matematis-statik</i>
NB	<i>Matematis-statik</i>
LogR	<i>Matematis-statik</i>
J48	<i>Tree-based Model</i>
DT	<i>Tree-based Model</i>

Maka, J48 cocok untuk digunakan pada data kelas biner berisi informasi numerik pada tiap atribut untuk jenis perawatan *container crane*. Gambar 6 menunjukkan ilustrasi sederhana J48 dengan contoh data dari dataset IRIS [26]. Keseluruhan algoritma ML terdaftar dalam Tabel 4. Algoritma RF, SVM, J48, dan DT termasuk ke dalam algoritma.

### C. Receiver Operating Characteristics

Dalam eksperimen ini, kami menggunakan *Receiver Operating Characteristics* (ROC) dan areanya (dinotasikan sebagai *Area Under Operating Characteristics* atau AUC) sebagai evaluasi sederhana mengenai bagaimana performa setiap algoritma ML mengklasifikasi per kelas perawatan ringan dan berat [28]. AUC juga digunakan sebagai alternatif dari metode evaluasi umum *Confusion Matrix*. Oleh karena WEKA secara eksklusif menggambarkan AUC per metode namun tidak dengan *Confusion Matrix* secara keseluruhan, kami menggunakan AUC.



Gambar 7. Ilustrasi ROC-AUC [28]

ROC seringkali dipasangkan dengan AUC. Kurva AUC-ROC merupakan pengukuran performa untuk masalah klasifikasi pada berbagai pengaturan *threshold*. ROC adalah kurva probabilitas dan AUC mewakili derajat atau ukuran keterpisahan. AUC memberi tahu

seberapa besar model mampu membedakan antar kelas. Semakin besar angka AUC, maka semakin baik metode tersebut dalam mengkararakteristikan kelas biner 1 dan 0, yang mana dalam kasus ini, yaitu jenis kelas perawatan sebagai ringan atau berat [29].

Model yang sangat baik memiliki AUC mendekati angka 1 yang berarti model tersebut memiliki ukuran keterpisahan yang baik. Model yang buruk memiliki AUC mendekati 0 yang berarti model tersebut memiliki ukuran keterpisahan yang terburuk. Jika nilai AUC pada suatu model adalah 0,5, maka, model tersebut tidak memiliki kapasitas pemisahan kelas sama sekali [25].

$$TPR = TP / TP + FN \quad (6)$$

$$Specificity = TN / TN + FP \quad (7)$$

$$FPR = 1 - Specificity = FP / TN + FP \quad (8)$$

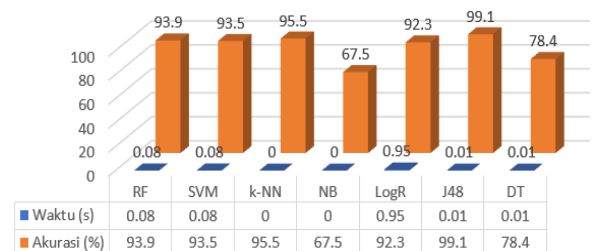
Gambar 7 menunjukkan ilustrasi bagaimana ROC dan AUC bekerja, dimana TPR adalah *True Positive Rate*, dan FPR adalah *False Positive Rate*. Persamaan (6) adalah kalkulasi TPR, Persamaan (7) menunjukkan kalkulasi *specificity* (hubungan antara TPR dan FPR), serta Persamaan (8) menunjukkan kalkulasi FPR.

### III. HASIL DAN PEMBAHASAN

Pada bagian ini, dijelaskan hasil uji data dengan komparasi pada setiap metode. Sebelum data dimasukkan untuk klasifikasi, kami menggunakan fungsi “*Replace Missing Values*” atau mengganti *value* yang hilang. Nilai yang hilang adalah jumlah (persentase) instance dalam data yang atributnya tidak ditentukan. Untuk menangani nilai yang hilang, kami menggunakan *filter* yang disebut “*Replace Missing Values*” untuk semua atribut nominal dan numerik dalam kumpulan data. Fungsi ini sudah *built-in* dalam aplikasi WEKA [30].

Tabel 5. *Datatable* Hasil Akurasi per Metode

Algoritma ML	Waktu (dalam detik)	Akurasi (%)
RF	0,08	93,9
SVM	0,08	93,5
k-NN	0	95,5
NB	0	67,5
LogR	0,95	92,3
J48	0,01	99,1
DT	0,01	78,4

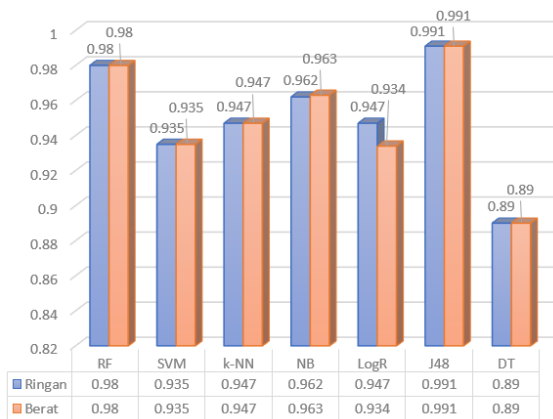


Gambar 8. Perbandingan Hasil Akurasi Klasifikasi per Algoritma

Ringkasan dari waktu eksekusi dan hasil akurasi ditunjukkan pada Gambar 8. Menggunakan WEKA,

kami berhasil mengklasifikasi jenis kelas perawatan *container crane* berdasarkan kelas biner: ringan dan berat. Ditunjukkan pada Tabel 5 dan Gambar 8, algoritma klasifikasi J48 memiliki akurasi tertinggi dengan angka 99,1%.

Gambar 9 menunjukkan angka AUC dari setiap algoritma ML. Berdasarkan Gambar 9, J48 kembali menunjukkan performa keterpisahan paling baik dengan nilai 99,1%. Tidak ada algoritma yang gagal mengenali karakteristik setiap kelas ringan atau berat: masing-masing memiliki nilai AUC diatas 0,5.



Gambar 9. Perbandingan Hasil AUC per Algoritma

#### IV. KESIMPULAN

*Container crane* memerlukan perawatan khusus dalam jangka waktu tertentu. Namun dalam prakteknya, sulit mengetahui kelas perawatan yang diperlukan walaupun data memadai. Algoritma ML mampu mengatasi masalah ini, dengan *task* klasifikasi sederhana berdasarkan data yang ada. Pengujian data *container crane* pada penelitian ini berfungsi untuk menentukan jenis kelas perawatan ringan atau berat pada *container crane*.

Dalam eksperimen ini, kami menggunakan algoritma matematis statistik dan algoritma berbasis DT. Algoritma tersebut adalah *Random Forest*, *Support Vector Machine*, *k-Nearest Neighbor*, *Naïve Bayes*, *Logistic Regression*, *J48*, dan *Decision Tree*. Algoritma ML ini kami gunakan pada perangkat lunak WEKA. Algoritma pengklasifikasi terbaik diraih oleh J48, dengan nilai akurasi 99,1%. Angka *Area Under the Receiver Operating Characteristics* milik J48 juga menunjukkan angka terbesar 99,1%, untuk setiap kelas ringan dan berat. Hasil tersebut berarti J48 memiliki performa tinggi dalam mengklasifikasi di masing-masing kelas ringan dan berat. Algoritma ML berbasis *tree-based model* ternyata memiliki keunggulan hasil klasifikasi dengan rata-rata nilai akurasi 91,2%, mengalahkan algoritma ML matematis-statistik yang memiliki nilai rata-rata akurasi 85,1%.

Sayangnya, NB menunjukkan performa dibawah rata-rata. Hasil ini mengindikasikan bahwa NB dipakai dengan *Entropy* tidak menunjukkan performa yang baik untuk data dengan karakteristik numerik-kategoris. Terlebih, kami menggunakan empat atribut dari total 10 atribut dari data *container crane*. Pada penelitian selanjutnya,

perlu dipertimbangkan penggunaan seluruh atribut namun data harus memiliki *value* yang komplit. Mengganti missing *value* dirasa penting, namun terlebih penting untuk memastikan bahwa data pada setiap atribut memiliki *value* yang *real*.

#### REFERENSI

- [1] L. Barua, B. Zou, and Y. Zhou, "Machine learning for international freight transportation management: A comprehensive review," *Res. Transp. Bus. Manag.*, vol. 34, p. 100453, Mar. 2020, doi: 10.1016/j.rtbm.2020.100453.
- [2] J. Malisan, I. Sadjiono, and B. Istijab, "Potensi Pusat Konsolidasi Pengangkutan untuk Regional Asia Pasifik: Suatu Analisis Kinerja," *J. Penelit. Transp. Laut*, vol. 20, no. 2, pp. 98–108, 2018.
- [3] A. Maddeppungeng and I. Suryani, "Optimalisasi komposisi alat berat pada proyek pembangunan pelabuhan pt. cemindo gemilang," *J. Konstr.*, vol. 9, no. 1, pp. 59–67, 2017.
- [4] X. Chen, R. Bai, R. Qu, H. Dong, and J. Chen, "A Data-Driven Genetic Programming Heuristic for Real-World Dynamic Seaport Container Terminal Truck Dispatching," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2020, pp. 1–8, doi: 10.1109/CEC48606.2020.9185659.
- [5] A. Krisdiyantoro, W. Wibawati, and H. Haryono, "Analisis Reliabilitas pada Hydraulic Spreader System Container Crane di PT. Terminal Petikemas Surabaya," *J. Sains dan Seni ITS*, vol. 8, no. 2, pp. D65–D71, 2020.
- [6] A. Khusyairi and E. S. Hisyam, "Analisis Kinerja Pelayanan Operasional Peti Kemas di Pelabuhan Pangkalbala Kota Pangkalpinang," in *FROPIL (Forum Profesional Teknik Sipil)*, 2016, vol. 4, no. 2, pp. 74–86.
- [7] L. Heilig, R. Stahlbock, and S. Voß, "From digitalization to data-driven decision making in container terminals," in *Handbook of Terminal Planning*, Springer, 2020, pp. 125–154.
- [8] M. Cannas, P. Fadda, G. Fancello, L. Frigau, and F. Mola, "Delay prediction in container terminals: A comparison of machine learning methods," in *13th world conference on transportation research*, 2013, pp. 1–6.
- [9] A. Viellechner and S. Spinler, "Novel Data Analytics Meets Conventional Container Shipping: Predicting Delays by Comparing Various Machine Learning Algorithms," 2020, doi: 10.24251/HICSS.2020.158.
- [10] D. Ignatov and A. Ignatov, "Decision Stream: Cultivating Deep Decision Trees," in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov. 2017, pp. 905–912, doi: 10.1109/ICTAI.2017.00140.
- [11] A. S. Dovbysh, V. V. Moskalenko, and A. S. Rizhova, "Information-Extreme Method for Classification of Observations with Categorical Attributes," *Cybern. Syst. Anal.*, vol. 52, no. 2, pp. 224–231, Mar. 2016, doi: 10.1007/s10559-016-9818-1.
- [12] H. Zareiforouh, S. Minaei, M. R. Alizadeh, and A. Banakar, "Qualitative classification of milled rice grains using computer vision and metaheuristic techniques," *J. Food Sci. Technol.*, vol. 53, no. 1, pp. 118–131, Jan. 2016, doi: 10.1007/s13197-015-1947-4.
- [13] Y. Prityanto and I. Pratama, "Missing Values Estimation on Multivariate Dataset: Comparison of Three Type

- Methods Approach,” in *2019 International Conference on Information and Communications Technology (ICOIACT)*, Jul. 2019, pp. 342–347, doi: 10.1109/ICOIACT46704.2019.8938472.
- [14] E. Frank and A. Mark, “Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for” Data Mining: Practical Machine Learning Tools and Techniques.” morgan kaufmann, 2016.
- [15] R. Couronné, P. Probst, and A.-L. Boulesteix, “Random forest versus logistic regression: a large-scale benchmark experiment,” *BMC Bioinformatics*, vol. 19, no. 1, p. 270, Dec. 2018, doi: 10.1186/s12859-018-2264-5.
- [16] H. Tran, “A Survey of Machine Learning and Data Mining Techniques used in Multimedia System,” 2019.
- [17] O. Oyeboade and D. E. Ighravwe, “Urban water demand forecasting: a comparative evaluation of conventional and soft computing techniques,” *Resources*, vol. 8, no. 3, p. 156, 2019.
- [18] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, “Efficient kNN classification with different numbers of nearest neighbors,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 5, pp. 1774–1785, 2018, doi: 10.1109/TNNLS.2017.2673241.
- [19] A. Navlani, “KNN Classification using Scikit-learn,” *Tutorials*, 2018. <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn> (accessed Jul. 01, 2020).
- [20] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, “Efficient kNN classification algorithm for big data,” *Neurocomputing*, vol. 195, pp. 143–148, 2016, doi: 10.1016/j.neucom.2015.08.112.
- [21] A. Faisal, Y. Alkhalifi, A. Rifai, and W. Gata, “Analisis Sentimen Dewan Perwakilan Rakyat Dengan Algoritma Klasifikasi Berbasis Particle Swarm Optimization,” *JOINTECS (Journal Inf. Technol. Comput. Sci.)*, vol. 5, no. 2, pp. 61–70, 2020.
- [22] and J. C. B. Pranali, B. Kanchan, “Emotion Recognition from Audio Signals using SVM and Naive Bayes : Review,” *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 7, no. 10, pp. 870–874, 2019.
- [23] D. J. and J. H. Martin, “Logistic Regression,” 2019.
- [24] M. A. Saif, A. N. Medvedev, M. A. Medvedev, and T. Atanasova, “Classification of Online Toxic Comments using the Logistic Regression and Neural Networks Models,” in *AIP Conference Proceedings*, 2018, vol. 2048, no. 1, p. 60011.
- [25] T. Pranckevičius and V. Marcinkevičius, “Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification,” *Balt. J. Mod. Comput.*, vol. 5, no. 2, p. 221, 2017.
- [26] K. Thirunavukkarasu, A. S. Singh, P. Rai, and S. Gupta, “Classification of IRIS Dataset using Classification Based KNN Algorithm in Supervised Learning,” in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, Dec. 2018, pp. 1–4, doi: 10.1109/CCAA.2018.8777643.
- [27] N. Saravana and D. V. Gayathri, “Performance and Classification Evaluation of J48 Algorithm and Kendall’s Based J48 Algorithm (KNJ48),” *Int. J. Comput. Trends Technol. (IJCTT)--Volume*, vol. 59, 2018.
- [28] Y. Liu and J. Shi, “Predicting the Potential Global Geographical Distribution of Two *Icerya* Species under Climate Change,” *Forests*, vol. 11, no. 6, p. 684, 2020.
- [29] S. Narkhede, “Understanding AUC-ROC Curve,” *Towar. Data Sci.*, vol. 26, 2018.
- [30] A. Desai and S. Rai, “Analysis of Machine Learning Algorithms using WEKA,” *Int. J. Comput. Appl.*, vol. 975, p. 8887, 2012.