PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika
Vol. 16, No. 1, Maret 2023, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v16i1.1982

Klasifikasi Insiden Pada Aplikasi Servicenow Untuk Menentukan Keberhasilan Service Level Agreement

Jemi Ardi Rawung[1)]; Achmad Solichin[1]*

1. Magister Ilmu Komputer, Universitas Budi Luhur, DKI Jakarta 12260, Indonesia

[*)]Email: 2011600950@student.budiluhur.ac.id

Received: 17 Maret 2023 | Accepted: 17 Maret 2023 | Published: 25 April 2023

ABSTRACT

The failure to achieve Service Level Agreement (SLA) has a negative impact on the satisfaction of the service that was promised. The SLA report that does not reach the target can have an impact on the promised service being not in accordance with what is paid by the business. On the other side, the service provider will have a bad reputation as a service provider company, and can even be fined for this. This study analyzed incidents using the CRISP-DM methodology and classification techniques with the Naive Bayes (NB), Logistic Regression (LR) and Support Vector Machine (SVM) algorithms which are most widely used in previous studies to predict SLA failures based on a review of studies. With help of the machine learning applications to do the data mining process, it is obtained that the SVM algorithm has the highest accuracy with a value of 75.05%, compared to LR 74.79% and NB 74.48%. The prediction prototypes are made with a programming language by adding incidents dataset, performing data reduction to balance data on target attribute, transforming data, and split training and testing dataset. This processes increasing the level of accuracy on the prototype to 83.32%.

Keywords: *SLA, Service, Incident, Clasification, Prediction*

ABSTRAK

Kegagalan pencapaian Service Level Agreement (SLA) berdampak buruk pada kepuasan sebuah layanan yang dijanjikan. Pelaporan SLA yang tidak mencapai target dapat berdampak pada tidak sesuainya layanan yang dijanjikan dengan apa yang dibayarkan oleh bisnis. Di sisi lain pihak yang memberikan layanan akan memiliki reputasi buruk sebagai perusahaan penyedia layanan, bahkan bisa sampai dikenakan denda untuk hal ini. Penelitian ini akan melakukan analisa insiden menggunakan metodologi CRISP-DM dan teknik klasifikasi dengan algoritma Naive Bayes (NB), Logistic Regression (LR) dan Support Vector Machine (SVM) yang paling banyak digunakan pada penelitian sebelumnya untuk melakukan prediksi kegagalan SLA berdasarkan pada tinjau studi. Dengan bantuan aplikasi machine learning untuk melakukan proses data mining, didapat algoritma SVM mempunyai akurasi tertinggi dengan nilai 75,05%, lalu LR 74,79% dan NB 74,48%. Prototipe prediksi dibuat menggunakan bahasa pemrograman dengan menambahkan dataset insiden, melakukan reduksi data guna menyeimbangkan data pada atribut target, transformasi data dan pemisahan antara data latihan dan tes. Proses ini menghasilkan tingkat akurasi pada prototipe bertambah menjadi 83,32%.

Kata kunci: *SLA, Layanan, Insiden, Klasifikasi, Prediksi*

Vol. 16, No. 1, Maret 2023, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v16i1.1982

1. PENDAHULUAN

Layanan Teknologi Informasi adalah suatu layanan yang disediakan oleh departemen Teknologi Informasi kepada para pengguna komputer atau perusahaan yang membutuhkan layanan Informasi Teknologi. Untuk menentukan kebutuhan dari layanan ini dibutuhkan persetujuan antara pemberi layanan dan penerima layanan. Sebuah kontrak yang mendefinisikan layanan yang diberikan, merinci dengan perjanjian, persyaratan yang digunakan, pihak yang terlibat, dan cara di mana ketidaksepakatan atau perubahan dinegosiasikan disebut Perjanjian Tingkat Lanjut atau *Service Level Agreement* (SLA) [1].

Sebagai contoh sebuah perusahaan *Fast Moving Consumer Goods* (FMCG) di Indonesia akan terus-menerus berkembang dari waktu ke waktu, menuntut kebutuhan teknologi baru yang menyebabkan banyak proyek yang berjalan untuk memenuhi kebutuhan perusahaan. Tuntutan kualitas layanan dari departemen Teknologi Informasi (TI) sangat dibutuhkan untuk menunjang kebutuhan tersebut. Optimalisasi kualitas penyampaian layanan terus menjadi pendorong penting bagi pertumbuhan bisnis. Permasalahan sering kali timbul dari perangkat keras komputer seperti jaringan infrastruktur, komponen komputer dan aksesorinya, selain itu di sisi aplikasi seperti sistem operasi, instalasi perangkat lunak maupun aplikasi pendukung untuk melakukan pekerjaan pengguna komputer sebagai karyawan suatu perusahaan.

Dalam rangka mengoptimalkan kualitas layanan salah satu upaya adalah memastikan semua masalah yang timbul dan menyebabkan layanan terganggu atau terhenti dapat diselesaikan sesuai dengan SLA yang sudah ditetapkan. Umumnya SLA terdapat 2 target yaitu, *Response Time*, yaitu ketika laporan suatu insiden ditanggapi, dan *Target Resolution*, yaitu waktu yang dibutuhkan untuk menyelesaikan insiden tersebut. Insiden adalah gangguan yang tidak direncanakan pada layanan atau penurunan kualitas layanan [2]. Semakin banyak permasalahan yang timbul terselesaikan tidak sesuai dengan SLA mengakibatkan kualitas pelayanan yang tidak baik, disisi lain untuk penyedia layanan tidak tercapainya SLA bisa berdampak negatif pada nama perusahaan penyedia layanan tersebut, bahkan bisa sampai dikenakan denda jika SLA yang dijanjikan tidak tercapai. Definisi lain menetapkan bahwa SLA adalah perjanjian negosiasi formal antara penyedia layanan dan pelanggan, yang dirancang untuk menciptakan pemahaman bersama tentang kualitas layanan, prioritas, dan tanggung jawab [3].

Perhitungan SLA dimulai ketika pembuatan tiket suatu insiden dilaporkan oleh pengguna dan ini menjadi acuan untuk menangani masalah tersebut. Sebagian ahli menyatakan bahwa Data Mining adalah langkah analisis terhadap proses penemuan pengetahuan di dalam basis data atau *Knowledge Discovery in Database* yang disingkat KDD [4]. Guna mengetahui dari kualitas pelayanan yang diberikan oleh departemen TI, metode Data Mining digunakan sehingga diharapkan dapat melihat seberapa jauh layanan departemen TI yang diberikan mengenai sasaran, mendukung kegiatan bisnis, dan dapat menentukan rencana untuk perbaikan serta pengembangan dari pelayanan yang diberikan sehingga dapat mengimbangi dan memenuhi kebutuhan dari suatu perusahaan.

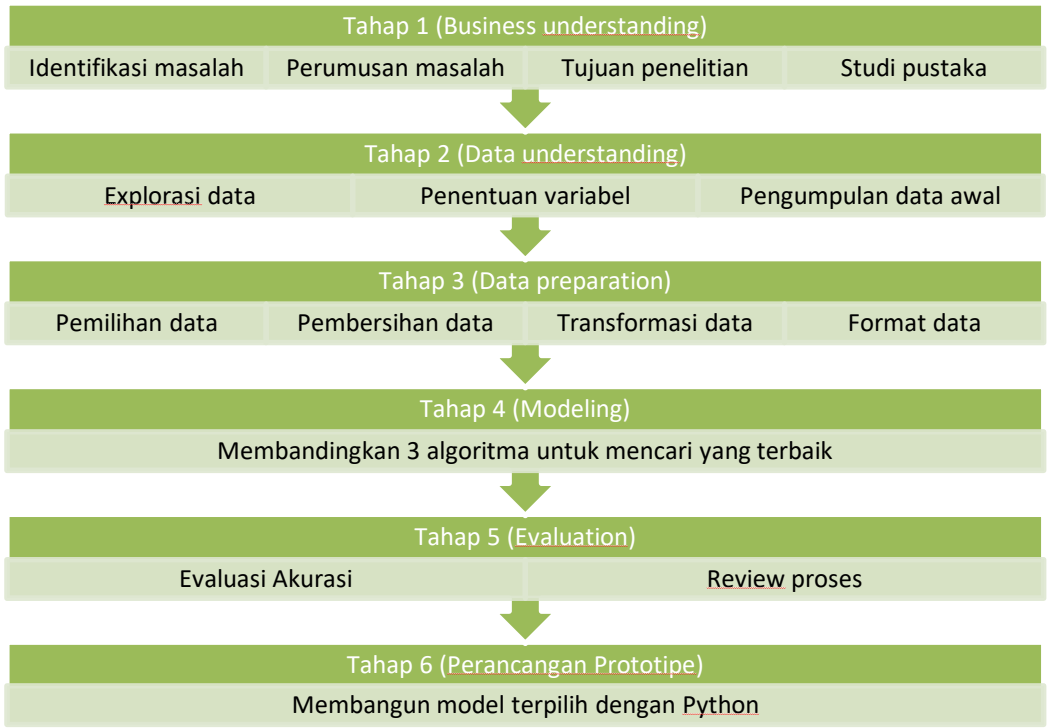
Terlihat di beberapa penelitian Data Mining yang menjadikan insiden sebagai objeknya dan menggunakan algoritma klasifikasi yang berbeda-beda. Ada yang mendapatkan hasil penelitian akurasi rata-rata pada Logistic Regression lebih tinggi yaitu 95.2% dibandingkan dengan Random Forest yang hanya 87% [5]. SVM dengan Kernel RBF bersama dengan 16 kategori *response time* untuk pelanggaran *cloud* QoS pada waktu *respons* dan *throughput* adalah model yang optimal. Model pembelajaran mesin yang diusulkan ini digabungkan dengan 16 kategori *response time* sehingga berkontribusi dalam memungkinkan penyedia layanan *cloud* untuk memprediksi terjadinya pelanggaran layanan berdasarkan waktu *respons* dan *throughput* [6]. Membandingkan tiga metode yang digunakan dengan komposisi dataset 70% sebagai *training set* dan 30% sebagai *testing set*.

Vol. 16, No. 1, Maret 2023, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v16i1.1982

Disimpulkan bahwa Random Forest adalah metode yang memberikan akurasi kinerja tertinggi [7]. Penggunaan algoritma Random Forest adalah performa terbaik karena kalsifikasi *tree based* kurang sensitif terhadap distribusi kelas [8]. Menggunakan 2 parameter yaitu, *Troughput* dan *Response time* akurasi prediksi dengan menggunakan algoritma SVM menunjukkan akurasi di atas 80% [9]. Dengan menggunakan aplikasi *Machine Learning* untuk membangun algoritma Regression untuk menentukan prediksi SLA, diperoleh akurasi 92% dengan tingkat kesalahan 8% [10]. Naive Bayes terbukti menjadi pendekatan yang efektif untuk memprediksi tingkat kepercayaan penyedia layanan untuk mencegah pelanggaran, sambil mempertimbangkan ketidakpastian lingkungan *cloud service* dengan prediksi rata-rata 95% [11]. Pada penelitian selanjutnya menggunakan algoritma Bayesian network untuk memprediksi SLA dan dilakukan 3 kali percobaan dengan hasil yang berbeda-beda, akan tetapi hasil akurasi tertinggi sampai 95% [12]. Pada penelitian lain yang menggunakan algoritma Bayesian memberikan kesimpulan bahwa dengan pelacakan skala besar Google, algoritma Bayes yang dirancang mengungguli solusi lain sebesar 5,6-50% dalam prediksi jangka panjang [13].

2. METODE

Ada beberapa metode pemrosesan dalam data mining di antaranya ada *Knowledge Discovery in Data* (KDD), *Sample, Explore, Modify, Model, and Assess* (SEMMA), *Data Science Process*, dan *Cross-Industry Standard Process for Data Mining* atau yang lebih dikenal dengan CRISP-DM [14]. Menurut [15] menggunakan CRISP-DM sebagai referensi model data mining proyek dimulai dengan mengerti bagaimana akan diterapkan. Metode yang paling cocok digunakan pada penelitian ini adalah *Cross-Industry Standard Process for Data Mining* atau CRISP-DM yang mempunyai 6 tahapan [16], dan dapat dilihat pada gambar 1.

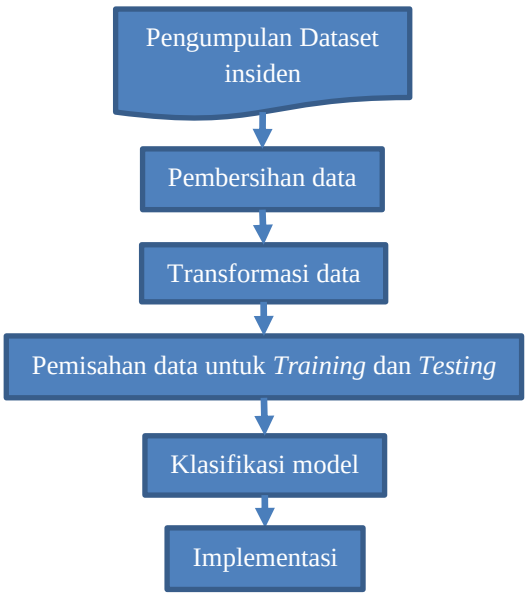


Gambar 1. Metode Penelitian

Untuk menentukan algoritma mana yang memiliki akurasi tertinggi dari beberapa penerapan model algoritma pada penelitian sebelumnya dan cocok dengan dataset penelitian ini, akan

Vol. 16, No. 1, Maret 2023, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v16i1.1982

menggunakan alat bantu aplikasi *Machine Learning* yang dapat menemukan algoritma prediksi terbaik untuk digunakan menjadi prototipe dengan bahasa pemrograman guna memprediksi keberhasilan SLA insiden yang baru. Sedangkan alur penelitian dapat dilihat pada gambar 2.



Gambar 2. Alur model klasifikasi

2.1. Pengumpulan Dataset Insiden

Pengambilan data dari aplikasi *Informasi Technology Service Management* (ITSM) sesuai dengan periode yang akan diteliti dan menerapkan beberapa filter untuk menyaring data yang akan diambil. Tabel yang akan digunakan adalah tabel insiden pada aplikasi ITSM, dan beberapa filter yang digunakan untuk membatasi data yang diambil seperti *Domain*, *Opened* atau *Created*. *Caller(caller_id)* diisi dengan tidak termasuk *EvAgent* yang berfungsi untuk menghindari insiden yang terbentuk karena log atau otomatis terbuat dikarenakan pesan *error* atau *alert* dari sistem.

2.2. Pembersihan Data

Proses pembersihan data berguna untuk mendapatkan data yang bersih karena kualitas sebuah dataset sangat berpengaruh terhadap kualitas model. Dengan menghilangkan data yang tidak bisa dilengkapi atau kosong, membuang data yang ganda, membuang data yang salah, dan juga ada beberapa insiden yang dibuka untuk kebutuhan testing dari sistem ketika ada perubahan sistem.

Dengan menggunakan algoritma Naive Bayes akan dilakukan percobaan sebanyak tiga kali dengan menggunakan pembagian dataset latihan dan tes yang telah ditentukan. Setelah itu dilanjutkan dengan algoritma Logistic Regression sebanyak tiga kali, lalu algoritma Support Vector Machine juga dilakukan sebanyak tiga kali percobaan.

2.3. Transformasi Data

Tahap selanjutnya adalah mengubah data untuk bisa digunakan seperti pada perubahan format data ke dalam bentuk file berformat ARFF, sehingga bisa diproses oleh aplikasi *machine learning*. Ketika data tidak seimbang pada aplikasi *machine learning* dapat menyeimbangkannya dengan memilih proses yang diinginkan, akan tetapi pada aplikasi pemrograman bisa menggunakan pemodelan *under sampling*.

Vol. 16, No. 1, Maret 2023, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v16i1.1982

2.4. Pemisahan data untuk *training* dan *test*.

Pembagian data yang digunakan untuk percobaan akan dilakukan dalam beberapa pembagian, di antaranya:

1. 70:30 (data yang digunakan untuk *training* atau percobaan adalah 70% dari total dataset sedangkan 30% digunakan untuk validasi [7][6].
2. 75:25 (data yang digunakan untuk *training* atau percobaan adalah 75% dari total dataset sedangkan 25% digunakan untuk validasi [9][11].
3. 80:20 (data yang digunakan untuk *training* atau percobaan adalah 80% dari total dataset sedangkan 20% digunakan untuk validasi [5].

2.5. Klasifikasi Model

Analisa dengan Data Mining menggunakan teknik klasifikasi dapat melakukan prediksi tingkat keberhasilan suatu SLA dari sebuah insiden yang timbul dan dengan membandingkan 3 algoritma yaitu Logistic Regression yang terutama digunakan untuk memodelkan variabel biner (0,1) berdasarkan satu atau lebih variabel lain yang disebut prediktif. Variabel biner yang dimodelkan umumnya disebut sebagai variabel respons atau variabel dependen [17].

Suatu metode di bidang statistik yang "dipinjam" oleh *machine learning* untuk diterapkan dalam komputer. Seperti namanya, inti dari metode ini adalah logistic fungtion yang memiliki keluaran berupa kurva berbentuk seperti huruf S, di mana nilanya antara 0 dan 1 [18]. Secara matematis, logistic fungtion ditulis seperti persamaan 1.

$$P(x) = \frac{1}{1+e^{-x}} \tag{1}$$

dengan e adalah konstanta Euler (nilainya sekitar 2,71828).

Metode Naive Bayes menggunakan teori Bayes yang ditemukan oleh Thomas Bayes seorang ahli matematika dari Inggris pada abad ke-18, dan namanya dijadikan sebagai nama teori tersebut. Dikutip dari [19], teori Bayes, probabilitas atau peluang bersyarat dinyatakan menggunakan persamaan 2.

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \tag{2}$$

Di mana X adalah bukti, H adalah hipotesis, P(H|X) adalah probabilitas bahwa hipotesis H benar untuk bukti X atau dengan kata lain P(H|X) merupakan probabilitas posterior H dengan syarat X, P(X|H) adalah probabilitas bahwa bukti X benar untuk hipotesis H atau probabilitas posterior X dengan syarat H, P(H) adalah probabilitas prior hipotesis H, dan P(X) adalah probabilitas prior bukti X.

Support Vector Machine diperkenalkan oleh Vapnik pada tahun 1992 sebagai teknik klasifikasi yang efisien untuk masalah nonlinier, menurut [20] algoritma Support Vector Machine menggunakan metode sekuensial diilustrasikan sebagai berikut:

1. Initiallization, $\alpha i = 0$
   Hitung matrix $D_{ij} = y_i y_j (K(x_i,x_j)+\lambda^2$
2. Lakukan tiga langkah di bawah ini untuk $i = 1, 2, ..., l$
   a. $E_i = \sum_{j=1}^{l} \alpha_j D_{ij}$
   b. $\delta\alpha_i = \min\{\max[\gamma(1- E_1),-\alpha_i], C-\alpha_i\}$
   c. $\alpha_i = \alpha_i + \delta\alpha_i$

Vol. 16, No. 1, Maret 2023, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v16i1.1982

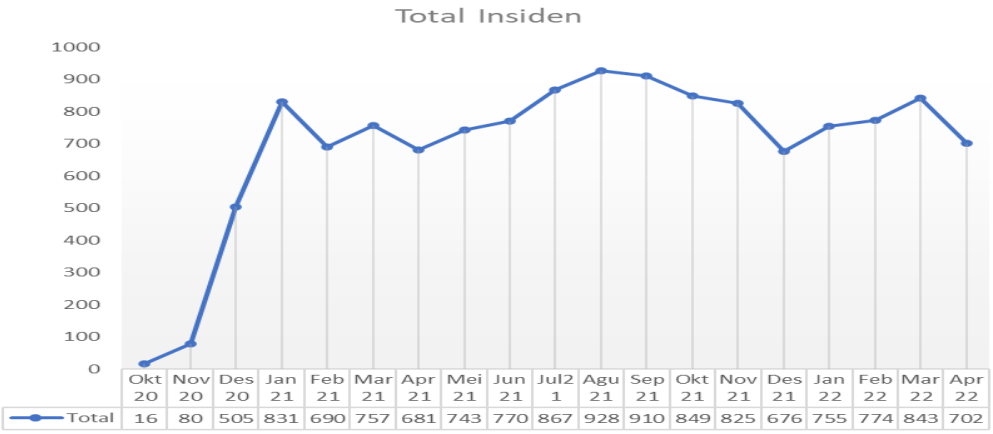
3. Kembali ke langkah 2 sampai α mencapai konvergenpenulisan gambar dan tabel berikut ini.

2.6. Implementasi

Langkah selanjutnya adalah pembuatan prototipe menggunakan algoritma terbaik yaitu SVM menggunakan bahasa pemrograman dengan tambahan dataset insiden untuk bulan Mei 2022 sampai dengan bulan Juni 2022. Jumlah data yang terkumpul setelah ditambahkan adalah 14560 insiden secara global untuk digunakan sebagai dataset yang baru.

3. HASIL DAN PEMBAHASAN

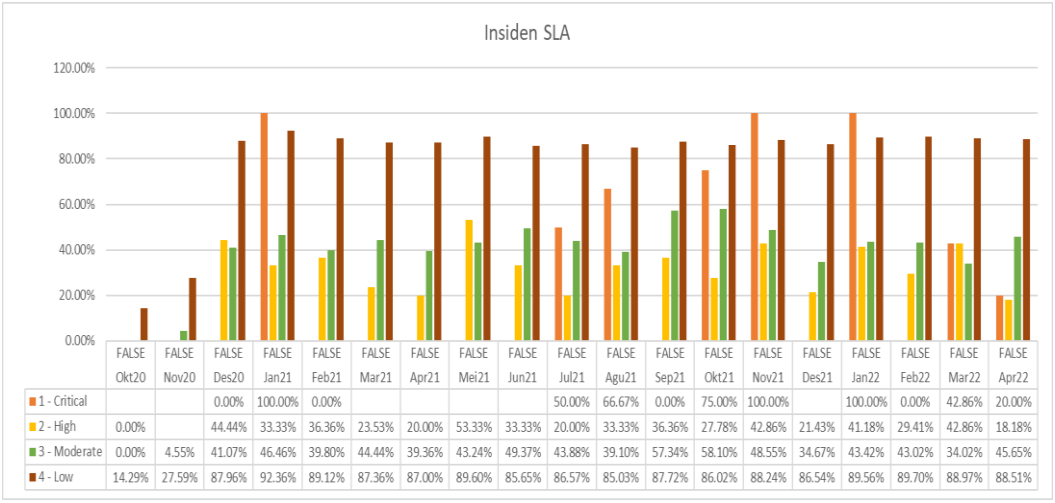
Dengan banyaknya insiden yang terselesaikan dengan status SLA yang gagal, dengan membuat sebuah *machine learning* dengan metode klasifikasi dan menerapkannya ketika sebuah insiden dilaporkan diharapkan bisa melakukan prediksi kegagalan insiden tersebut, dengan memberikan perhatian khusus untuk mencegah insiden tersebut gagal diselesaikan sesuai SLA yang diinginkan. Dengan menggunakan data insiden hasil penarikan dari aplikasi ITSM selama 19 bulan dimulai dari Oktober 2020 sampai dengan April 2022 dengan total 13202 insiden, terlihat volume insiden setiap bulan yang berbeda pada terlihat pada gambar 3.



Gambar 3. Jumlah insiden selama 19 bulan.

Pada 2 bulan pertama menunjukkan jumlah insiden yang tidak wajar karena jumlahnya sangat sedikit dibandingkan dengan bulan-bulan yang lain yang di sebabkan pada bulan tersebut terjadi transisi proses pemindahan data dari server *production* ke server *archive*. Ini disebabkan server *production* yang mengalami penurunan kecepatan ketika diakses dan dilakukan penarikan data. Sedangkan pencapaian SLA untuk insiden 19 bulan pada gambar 4, terlihat banyak yang di bawah target, hanya ada beberapa insiden dengan prioritas 1 yang SLA-nya tercapai, sisanya terlihat semua di bawah 95%.

Vol. 16, No. 1, Maret 2023, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v16i1.1982



	FALSE Okt20	FALSE Nov20	FALSE Des20	FALSE Jan21	FALSE Feb21	FALSE Mar21	FALSE Apr21	FALSE Mei21	FALSE Jun21	FALSE Jul21	FALSE Agu21	FALSE Sep21	FALSE Okt21	FALSE Nov21	FALSE Des21	FALSE Jan22	FALSE Feb22	FALSE Mar22	FALSE Apr22
1 - Critical			0.00%	100.00%	0.00%					50.00%	66.67%	0.00%	75.00%	100.00%		100.00%	0.00%	42.86%	20.00%
2 - High	0.00%		44.44%	33.33%	36.36%	23.53%	20.00%	53.33%	33.33%	20.00%	33.33%	36.36%	27.78%	42.86%	21.43%	41.18%	29.41%	42.86%	18.18%
3 - Moderate	0.00%	4.55%	41.07%	46.46%	39.80%	44.44%	39.36%	43.24%	49.37%	43.88%	39.10%	57.34%	58.10%	48.55%	34.67%	43.42%	43.02%	34.02%	45.65%
4 - Low	14.29%	27.59%	87.96%	92.36%	89.12%	87.36%	87.00%	89.60%	85.65%	86.57%	85.03%	87.72%	86.02%	88.24%	86.54%	89.56%	89.70%	88.97%	88.51%

Gambar 4. Grafik SLA selama 19 bulan.

Dalam pencatatan insiden pada aplikasi ITSM terdapat banyak atribut-atribut, akan tetapi pada penelitian ini hanya atribut yang penting yang digunakan seperti;

1. *Number*/Nomor, berisikan nomor urut dari insiden yang dicatat pada aplikasi setiap kali pelaporan terjadi.
2. *State*/Status, berisikan status terakhir dari sebuah insiden.
3. *Location*/Lokasi, yang menjelaskan lokasi pelapor insiden.
4. *Category*/Kategori, berisikan data kategori jenis permasalahan yang dilaporkan dalam sebuah insiden.
5. *Assignment Group*/Tim yang menyelesaikan, berisikan data tim mana yang akan menangani insiden.
6. *Priority*/Prioritas, berisikan informasi seberapa tinggi tingkat prioritas dari sebuah insiden yang dilaporkan.
7. *Created*/Dibuat, berisikan informasi yang menunjukkan waktu sebuah insiden dibuat, dan dilaporkan.
8. *Contact type*/Tipe kontak, berisikan informasi bagaimana sebuah insiden dilaporkan oleh pengguna.
9. *Closed*/Ditutup, berisikan informasi yang menunjukkan waktu sebuah insiden ditutup.
10. *Has breached*/Telah melanggar, berisikan informasi apakah sebuah insiden telah gagal memenuhi SLA.

Dataset sebanyak 13202 insiden yang di dapat akan dilakukan pembersihan sebelum digunakan pada *machine learning* dan berikut adalah beberapa proses yang dilakukan untuk pembersihan data;

1. Menghilangkan semua insiden yang tidak terdapat informasi lokasi karena insiden seperti ini adalah insiden yang terbentuk dari log atau otomatis ketika ada *error* atau *event* dari sebuah *task* di server.
2. Penambahan data kategori yang hilang atau yang kosong dengan memeriksa insiden pada aplikasi ITSM.
3. Menghilangkan data yang statusnya belum terselesaikan, karena jika insiden tersebut masih belum selesai maka perhitungan SLA juga belum terjadi.
4. Menghilangkan tiket dengan prioritas 5 karena tidak dihitung dalam SLA.

Vol. 16, No. 1, Maret 2023, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v16i1.1982

5. Menghilangkan insiden yang pada kolom tipe kontak dengan nilai sama dengan *Alert* dan *Event* karena itu merupakan insiden yang terbentuk otomatis karena *Event* ataupun *Alert*.
6. Menghilangkan insiden yang pada kolom *Description* kosong (insiden yang terbuat karena kesalahan) atau dimulai dengan *Event*.
7. Menghilangkan insiden tes atau hanya dibuat sebagai tes insiden, ataupun ketika ada masalah dengan aplikasi ITSM.
8. Menambahkan kolom *Country* dan *Code*, yang akan mengubah kolom lokasi menjadi numerik berdasarkan *Country Code.*
9. Menambahkan kolom baru untuk *Assignment group* dan melakukan pengelompokan terhadap tim yang menyelesaikan masalah berdasarkan level dari tim tersebut.
10. Untuk atribut *Priority* hanya diambil angkanya saja.
11. Untuk atribut *Breached* nilai *FALSE* dan *TRUE* diubah menjadi numerik 1 dan 0.

Setelah dataset sudah siap diproses menggunakan aplikasi *machine learning* dan dapat disimpulkan, untuk percobaan menggunakan algoritma Naive Bayes, terlihat pada bagian hasil, untuk *Correctly Classified Instances* atau hasil insiden yang diprediksi dengan benar sebanyak 1162 dengan persentase 74,1544%, sedangkan untuk *Incorrectly Classified Instances* atau hasil insiden yang diprediksi salah sebanyak 405 atau dengan persentase 25,8456%. Lebih detail lagi bisa dilihat pada bagian *Confusion Matrix*, di sini dapat dijelaskan bahwa terdapat 613 insiden termasuk dalam *True Positif* atau insiden dengan SLA yang tercapai dan diprediksi dengan benar, 186 insiden *False Positif* atau insiden dengan SLA yang tercapai dan diprediksi dengan salah, 219 insiden *False Negatif* atau insiden dengan SLA tidak tercapai dan diprediksi salah, sedangkan 549 insiden *True Negatif* atau insiden dengan SLA tidak tercapai dan diprediksi dengan benar.

Percobaan dengan menggunakan algoritma ini dilakukan sebanyak 3 kali dengan menggunakan pembagian dataset latihan dan tes. Setelah Naive bayes lalu dilanjutkan dengan algoritma Logistic Regression sebanyak 3 kali, lalu algoritma Support Vector Machine juga dilakukan sebanyak 3 kali percobaan dengan pembagian dataset latihan dan tes yang berbeda. Dari semua percobaan yang telah dilakukan pada setiap algoritma dan juga dilakukan berulang dengan dataset latihan dan tes yang berbeda maka dapat dihasilkan tingkat akurasi yang berbeda-beda seperti yang terlihat pada tabel 1. Terlihat akurasi tertinggi dengan nilai 74.7926% adalah algoritma Logistic Regression dan Support Vector Machine menggunakan pembagian dataset 70:30.

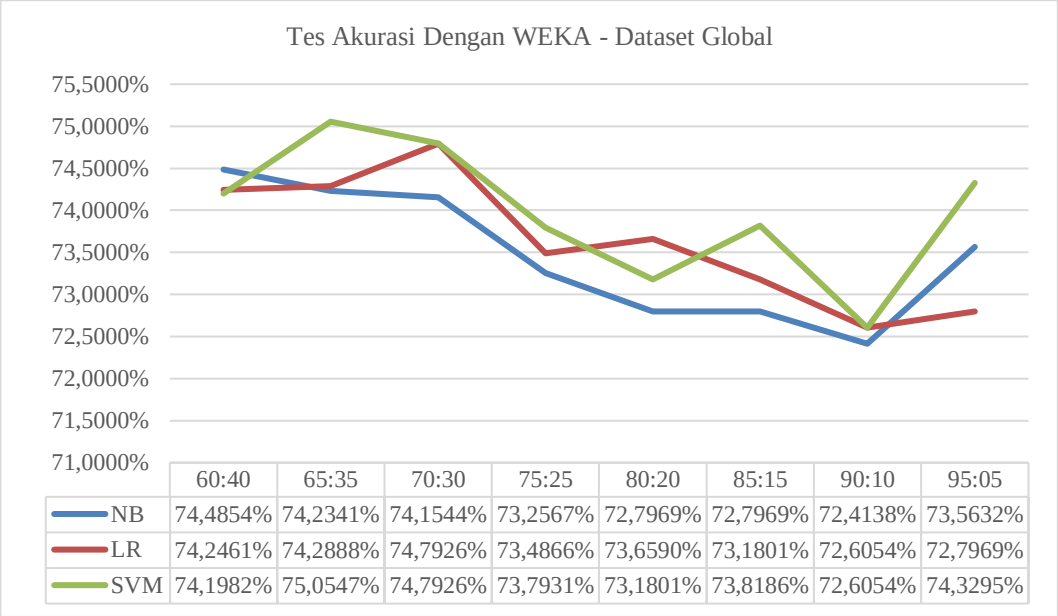
Tabel 1. Hasil akurasi.

	Split Traning		
	70:30	**75:25**	**80:20**
Naive Bayes (NB)	74.1544%	73.2567%	72.7969%
Logistic Regression (LR)	74.7926%	73.4866%	73.6590%
Support Vector Machine (SVM)	74.7926%	73.7931%	73.1801%

Dikarenakan mempunyai hasil akurasi yang imbang antara logistic Regression dan Support Vector Machine dengan menggunakan data latihan dan tes 70:30, maka dilakukan percobaan tambahan dengan menggunakan dataset latihan dan tes yang berbeda dari sebelumnya, yaitu 60:40, 65:35, 85:15, 90:10 dan 95:05. Dari percobaan yang dilakukan menggunakan aplikasi *machine learning* dapat dilihat pada gambar 5. Secara keseluruhan dari semua algoritma-algoritma dan pembagian data set yang berbeda, di sini terlihat algoritma yang terbaik adalah Suport Vector

Vol. 16, No. 1, Maret 2023, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v16i1.1982

Machine ketika dilakukan tes menggunakan pembagian dataset 65:35 dan akurasinya yaitu 75,0547%.



	60:40	65:35	70:30	75:25	80:20	85:15	90:10	95:05
NB	74,4854%	74,2341%	74,1544%	73,2567%	72,7969%	72,7969%	72,4138%	73,5632%
LR	74,2461%	74,2888%	74,7926%	73,4866%	73,6590%	73,1801%	72,6054%	72,7969%
SVM	74,1982%	75,0547%	74,7926%	73,7931%	73,1801%	73,8186%	72,6054%	74,3295%

Gambar 5. Hasil tes keseluruhan pada aplikasi *machine learning.*

Setelah mendapatkan algoritma terbaik yaitu SVM, langkah selanjutnya adalah membuat prototipe menggunakan bahasa pemrograman dengan algoritma SVM dengan perbandingan dataset *training* dan *test* adalah 65:35, lalu mengimplementasikannya dengan dataset sebanyak 14560 insiden yang sudah ditambahkan dari dataset sebelumnya. Dataset tersebut ternyata tidak seimbang antara atribut target yaitu *Has breached* bernilai *false* dengan yang *true*, untuk menyeimbangkan dataset ini maka digunakan teknik *undersampling* yang berakibat mengurangi dataset. Setelah melakukan proses modeling menggunakan algoritma SVM, hasilnya memperlihatkan akurasi model yang tunjukkan lebih tinggi yaitu 83,3203%.

Hasil *confusion matrix* yang dihasilkan bisa disimpulkan bahwa nilai *True Positif* yaitu insiden dengan SLA berhasil tercapai dan diprediksi secara benar sebanyak 3857 insiden, sedangkan *False Negatif* yaitu insiden dengan SLA yang gagal dan di prediksi SLA berhasil tercapai sebanyak 258 insiden, lalu untuk *True Negatif* yaitu insiden dengan SLA tidak berhasil tercapai dan diprediksi dengan benar sebanyak 389 insiden, sedangkan *False Positif* yaitu insiden yang SLA berhasil tercapai dan di prediksi SLA tidak berhasil tercapai sebanyak 592 insiden.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang telah dilakukan dengan menggunakan dataset insiden pada aplikasi ITSM yang di ambil sebanyak 19 bulan, dan di dalamnya terdapat enam buah atribut untuk dilakukan modeling menggunakan 3 algoritma Logistic Regression, NaiveBayes, dan Support Vector Machine dengan hasil akurasi yang imbang antara Logistic Regression, lalu dengan tambahan percobaan membuktikan bahwa Support Vector Machine lebih unggul daripada Logistic Regression pada dataset insiden yang digunakan ini.

Klasifikasi dengan algoritma Support Vector Machine yang diterapkan pada prototipe dapat digunakan sebagai alat bantu untuk memprediksi kegagalan SLA dari insiden yang dilaporkan pengguna. Jika suatu insiden dapat diketahui akan gagal maka tim TI bisa melakukan pencegahan

Vol. 16, No. 1, Maret 2023, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v16i1.1982

dengan cara memberikan perhatian khusus pada insiden ini atau melakukan pemantauan secara berkala dalam proses penyelesaian insiden tersebut. Secara otomatis prototipe ini dapat mengurangi insiden-insiden yang terselesaikan di luar dari target SLA akan tetapi akan menjadi 2 kali proses karena aplikasi ITSM dan prototipe adalah 2 aplikasi terpisah. Alangkah baiknya jika terdapat aplikasi ITSM yang berguna untuk pencatatan insiden mempunyai kemampuan memprediksi tingkat kegagalan dari sebuah insiden yang muncul.

Untuk dataset atribut *Assignment*, terlihat kurang mewakili dari grup yang menyelesaikan masalah, terlihat banyak insiden yang diselesaikan oleh level 1 yaitu Service Desk. Untuk penelitian lebih lanjut mungkin bisa digunakan atribut lain seperti waktu yang dibutuhkan dari perpindahan *Assignment* ke level grup yang berbeda, ataupun dilakukan proses tambahan data mining pada keterangan insiden atau *incident description.*

DAFTAR PUSTAKA

[1] A. Ahmad, R. A. Arshah, A. Kamaludin, and L. Ngah, "Adopting of Service Level Agreement (SLA) in enhancing the quality of IT hardware service support," 2020.

[2] AXELOS, *ITIL Foundation, ITIL (ITIL 4 Foundation)*, 4th editio. London: TSO (The Stationery Office), 2019.

[3] N. Ahmad and Z. M. Shamsudin, "Systematic approach to successful implementation of ITIL," *Procedia Comput. Sci.*, vol. 17, pp. 237–244, 2013, doi: 10.1016/j.procs.2013.05.032.

[4] T. Hendrickx, B. Cule, P. Meysman, S. Naulaerts, K. Laukens, and B. Goethals, "Mining association rules in graphs based on frequent cohesive itemsets," 2015. doi: 10.1007/978-3-319-18032-8_50.

[5] S. Gouryraj, S. Kataria, and J. Swvigaradoss, "Service Level Agreement Breach Prediction in ServiceNow," *Proc. 3rd Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2021*, pp. 689–698, 2021, doi: 10.1109/ICIRCA51532.2021.9544916.

[6] T.-S. Wong, G.-Y. Chan, and F.-F. Chua, "A Machine Learning Model for Detection and Prediction of Cloud Quality of Service Violation," O. Gervasi, B. Murgante, S. Misra, E. Stankova, C. M. Torre, A. M. A. C. Rocha, D. Taniar, B. O. Apduhan, E. Tarantino, and Y. Ryu, Eds. Cham: Springer International Publishing, 2018.

[7] R. Pasquini, F. Moradi, J. Ahmed, A. Johnsson, C. Flinta, and R. Stadler, "Predicting SLA conformance for cluster-based services," *2017 IFIP Netw. Conf. IFIP Netw. 2017 Work.*, vol. 2018-Janua, pp. 1–2, 2017, doi: 10.23919/IFIPNetworking.2017.8264873.

[8] R. A. Hemmat and A. Hafid, "SLA Violation Prediction In Cloud Computing: A Machine Learning Perspective," 2016.

[9] A. F. M. Hani, I. V. Paputungan, and M. F. Hassan, "Support Vector regression for Service Level Agreement violation prediction," *Proceeding - 2013 Int. Conf. Comput. Control. Informatics Its Appl. "Recent Challenges Comput. Control Informatics", IC3INA 2013*, pp. 307–311, 2013, doi: 10.1109/IC3INA.2013.6819192.

[10] P. Leitner, B. Wetzstein, F. Rosenberg, A. Michlmayr, S. Dustdar, and F. Leymann, "Runtime Prediction of Service Level Agreement Violations for Composite Services," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6275 LNCS, no. May 2014, 2010, doi: 10.1007/978-3-642-16132-2.

[11] O. Jules, A. Hafid, and M. A. Serhani, "Bayesian network, and probabilistic ontology driven trust model for SLA management of Cloud services," *2014 IEEE 3rd Int. Conf. Cloud Networking, CloudNet 2014*, no. October, pp. 77–83, 2014, doi: 10.1109/CloudNet.2014.6968972.

Vol. 16, No. 1, Maret 2023, P-ISSN 1978-9262, E-ISSN 2655-5018
DOI: https://doi.org/10.33322/petir.v16i1.1982

[12] S. Duan and S. Babu, "Proactive identification of performance problems," *Proc. ACM SIGMOD Int. Conf. Manag. Data*, pp. 766–768, 2006, doi: 10.1145/1142473.1142582.

[13] S. Di, D. Kondo, and W. Cirne, "Host load prediction in a Google compute cloud with a Bayesian model," *Int. Conf. High Perform. Comput. Networking, Storage Anal. SC*, 2012, doi: 10.1109/SC.2012.68.

[14] D. C. Corrales, A. Ledezma, and J. C. Corrales, "From theory to practice: A data quality framework for classification tasks," *Symmetry (Basel).*, vol. 10, no. 7, pp. 1–29, 2018, doi: 10.3390/sym10070248.

[15] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th Editio. 2016.

[16] C. Pete *et al.*, "CRISM-DM 1.0: Step-by-step data mining guide," 2000.

[17] J. Hilbe, *Practical Guide to Logistic Regression*, Illustrate. Boca Raton: CRC Press, 2016.

[18] D. Kurniawan, *Pengenalan Machine Learning dengan Python*, vol. 2. Jakarta: PT. Elex Media Komputindo, 2021.

[19] Suyanto, *Data Mining Untuk Klasifikasi dan Klasterisasi Data. Informatika Bandung*, *Bandung.*, Revise., vol. 1. Bandung: Informatika Bandung, 2017.

[20] S. Vijayakumar and S. Wu, "Sequential Support Vector Classi ers and Regression," 1999, vol. 619, pp. 610–619.