

Implementasi Algoritma Machine Learning untuk Pengelolaan Laporan pada Website Dinas Penanaman Modal Dan Pelayanan Terpadu Satu Pintu (DPMPTSP) Kota Surabaya

Naufal Iqbal Zaki¹, Anita Qoiriah²

^{1,2}Teknik Informatika/S-1 Teknik Informatika, Universitas Negeri Surabaya

¹naufal.21062@mhs.unesa.ac.id

²anitaqoiriah@unesa.ac.id

Abstrak— Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu (DPMPTSP) Kota Surabaya menghadapi tantangan dalam pengelolaan laporan perizinan akibat tingginya volume data dan proses manual yang kurang efisien. Penelitian ini mengimplementasikan algoritma *Machine Learning* dalam sistem pengelolaan laporan berbasis web guna mengotomatisasi proses klasifikasi dan prediksi secara cerdas. Algoritma *Naive Bayes* digunakan untuk memprediksi jenis layanan, sedangkan *Regresi Linear* digunakan untuk memprediksi durasi layanan. Dataset yang digunakan terdiri dari 3.434 entri laporan perizinan bulan November 2024 yang telah melalui proses *preprocessing*. Hasil pelatihan model *Naive Bayes* menunjukkan akurasi sebesar 95–96%, dengan presisi tertinggi 97% pada skenario pembagian data 70:30 serta *F1-Score* yang stabil pada angka 96%. Model *Regresi Linear* menunjukkan performa terbaik pada skenario 85:15 dengan *Mean Absolute Error* (MAE) sebesar 0,38 hari dan *Mean Absolute Percentage Error* (MAPE) sebesar 8,42%. Sistem ini terbukti mampu mengurangi kesalahan manual, mempercepat pemrosesan laporan, dan meningkatkan keandalan hasil prediksi. Antarmuka sistem yang interaktif dan mudah digunakan telah diuji oleh 18 responden, dengan tingkat kepuasan mencapai 100%. Hasil ini menunjukkan bahwa implementasi algoritma *Machine Learning* secara signifikan dapat meningkatkan efisiensi dan akurasi pelayanan publik, serta berpotensi menjadi model rujukan bagi pengembangan sistem serupa di instansi pemerintahan lainnya.

Kata Kunci— *Machine Learning, Naive Bayes, Regresi Linear, Laporan Perizinan, DPMPTSP Surabaya.*

I. PENDAHULUAN

Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu (DPMPTSP) Kota Surabaya memiliki peran strategis dalam penyelenggaraan layanan perizinan terpadu di wilayah Kota Surabaya. Lembaga ini dibentuk berdasarkan Peraturan Wali Kota Surabaya Nomor 084 Tahun 2021, yang mengatur kedudukan, struktur organisasi, tugas, fungsi, serta sistem kerjanya [1]. Seiring transformasi digital, pemanfaatan teknologi informasi menjadi aspek penting dalam meningkatkan efisiensi dan kualitas layanan publik. Sektor perizinan, sebagai bagian vital dari proses investasi dan pertumbuhan ekonomi, juga menuntut pelayanan yang cepat, akurat, dan transparan.

Namun, seiring meningkatnya volume pengajuan perizinan, DPMPTSP menghadapi tantangan signifikan dalam pengelolaan laporan, khususnya dalam proses klasifikasi dan

verifikasi yang masih dilakukan secara manual. Kondisi ini menyebabkan proses menjadi lambat, tidak efisien, serta berpotensi menimbulkan kesalahan. Oleh karena itu, diperlukan sistem cerdas yang mampu mengotomatisasi proses klasifikasi dan prediksi berdasarkan data historis [2].

Pendekatan berbasis *Machine Learning* menawarkan solusi potensial terhadap permasalahan tersebut. Teknologi ini memungkinkan sistem untuk mengenali pola dan melakukan prediksi secara otomatis dengan akurasi yang tinggi. Dalam konteks ini, algoritma *Naive Bayes* digunakan untuk mengklasifikasikan jenis layanan berdasarkan dokumen laporan, sedangkan *Regresi Linear* digunakan untuk memprediksi durasi pelayanan berdasarkan variabel-variabel seperti jenis perizinan, nama pemohon, dan waktu masuk berkas.

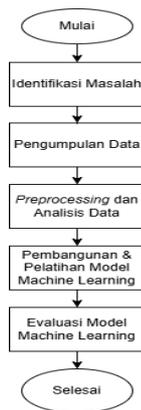
Naive Bayes merupakan metode klasifikasi probabilistik sederhana yang menganalisis frekuensi kemunculan serta kombinasi atribut dalam dataset, dengan asumsi independensi antar fitur terhadap variabel kelas [3]. Efektivitasnya telah dibuktikan oleh Saputra dan Soetanto [4], yang mengaplikasikan *Naive Bayes* untuk mengklasifikasikan wilayah terdampak banjir di Jakarta Pusat, dengan akurasi 97,73%, *recall* 100%, dan *F1-Score* 88,98%.

Di sisi lain, *Regresi Linear* merupakan metode statistik yang digunakan untuk memodelkan hubungan linier antara variabel dependen, seperti waktu penyelesaian layanan, dan beberapa variabel independen. Fawwazrin [5] berhasil menggunakan metode ini untuk memprediksi durasi pencetakan KTP di Disdukcapil Kecamatan Samarinda Utara dengan akurasi R^2 sebesar 98,38% dan MAE 11,86 menit.

Integrasi kedua algoritma tersebut ke dalam sistem berbasis web diharapkan dapat meningkatkan efisiensi pengolahan laporan, mengurangi kesalahan manual, dan mendukung pengambilan keputusan berbasis data di DPMPTSP Surabaya. Selain itu, sistem ini juga diharapkan dapat menjadi rujukan bagi instansi pemerintah lainnya dalam menerapkan solusi berbasis teknologi cerdas untuk meningkatkan kualitas pelayanan publik.

Penelitian ini tidak hanya mencakup pengembangan model dan sistem, tetapi juga menangani tantangan seperti kualitas data yang bervariasi dan kebutuhan adaptasi model terhadap perubahan pola pengajuan. Oleh karena itu, dilakukan proses *preprocessing* data, pelatihan, pengujian, dan evaluasi performa algoritma secara menyeluruh.

II. METODE PENELITIAN



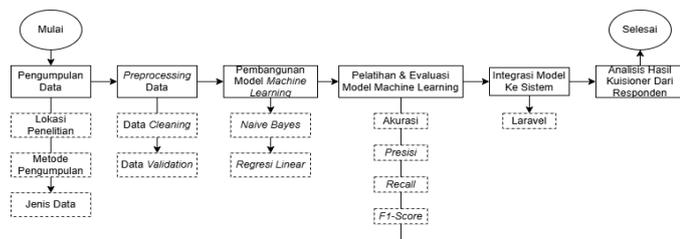
Gbr. 1 Diagram Alur Penelitian

Pada **Gbr. 1** Diagram Alur Penelitian, diagram menggambarkan tahapan penelitian yang dimulai dari identifikasi masalah, pengumpulan data, *preprocessing* data, pengembangan dan pelatihan model *Machine Learning*, implementasi sistem, hingga evaluasi hasil respon pengguna. Setiap langkah berfokus pada meningkatkan efisiensi pengelolaan laporan di DPMPTSP Kota Surabaya.

A. Identifikasi Masalah

Penelitian ini mengkaji permasalahan dalam sistem pengelolaan laporan di Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu (DPMPTSP) Kota Surabaya, khususnya terkait efisiensi pengelolaan, keterlambatan rekapitulasi bulanan, serta potensi kesalahan manusia (*human error*) dalam proses input data. Tujuan dari penelitian ini adalah untuk mengatasi permasalahan tersebut guna meningkatkan kualitas pelayanan dan mempercepat proses rekapitulasi laporan bulanan.

Sebagai solusi, penelitian ini mengembangkan sistem pengelolaan laporan berbasis *Machine Learning* yang mengintegrasikan dua algoritma utama yaitu *Naive Bayes*, yang digunakan untuk memprediksi jenis layanan berdasarkan data laporan, dan *Regresi Linear*, yang digunakan untuk memprediksi estimasi durasi layanan. Alur penelitian mencakup tahapan pengumpulan data, *preprocessing*, pembangunan dan pelatihan model *Machine Learning*, integrasi model ke dalam sistem berbasis *Laravel*, hingga evaluasi model dan analisis hasil tanggapan pengguna melalui kuisioner. Diagram alur riset secara lengkap ditampilkan pada **Gbr 2**.



Gbr. 2 Diagram Riset Penelitian

B. Pengumpulan Data

Data dalam penelitian ini diperoleh dari DPMPTSP Kota Surabaya, berupa file *Excel* yang memuat informasi perizinan seperti nomor berkas, tanggal pendaftaran, nama pemohon, jenis perizinan, dinas terkait, status, nama petugas, dan nomor telepon. Penelitian dilaksanakan langsung di kantor DPMPTSP Surabaya yang berlokasi di Jalan Jimerto No. 25–27, dan difokuskan pada pengembangan sistem pengelolaan laporan berbasis *Machine Learning* untuk meningkatkan efisiensi dan akurasi pengolahan berkas. Jenis data yang digunakan mencakup data struktural (terorganisir dalam format tabel) dan data tekstual (berupa catatan atau keterangan tambahan dari petugas). Metode pengumpulan data meliputi studi dokumentasi terhadap arsip laporan perizinan dan wawancara dengan staf DPMPTSP untuk menggali kebutuhan sistem, tantangan proses manual, serta validasi terkait format dan struktur data yang digunakan dalam pengembangan model klasifikasi (*Naive Bayes*) dan prediksi durasi layanan (*Regresi Linear*).

C. Metode Analisis Data

Analisis data dalam penelitian ini bertujuan untuk mengevaluasi kinerja sistem pengelolaan laporan yang dikembangkan menggunakan algoritma *Naive Bayes* dan *Regresi Linear*. Tahapan analisis mencakup persiapan data, penerapan model, dan evaluasi performa. Persiapan data meliputi pembersihan data tidak lengkap dan duplikat, transformasi format sesuai kebutuhan model, pemisahan data menjadi data latih dan data uji, serta normalisasi agar seluruh variabel berada pada skala yang seragam, khususnya untuk mendukung akurasi dalam model *Regresi Linear*. Setelah model dilatih dan diterapkan, kinerjanya dievaluasi menggunakan metrik akurasi, presisi, *recall*, dan *F1-score* untuk *Naive Bayes*, serta MAE (*Mean Absolute Error*) dan MAPE (*Mean Absolute Percentage Error*) untuk *Regresi Linear*.

D. Implementasi Machine Learning

Machine Learning merupakan bagian dari kecerdasan buatan (*Artificial Intelligence*) yang dirancang untuk memecahkan masalah secara efisien dan mempermudah pelaksanaan tugas tertentu [6]. Teknologi ini memungkinkan komputer belajar dari pengalaman melalui algoritma khusus, sehingga dapat mengambil keputusan tanpa pemrograman berulang [7]. Sebagai cabang dari ilmu komputer, *machine learning* meniru cara belajar manusia dan terus meningkatkan akurasi prediksi, dengan hasil yang sangat bergantung pada kecanggihan algoritma yang digunakan [8]. Salah satu implementasinya adalah dalam sistem manajemen laporan di DPMPTSP Kota Surabaya, yang memanfaatkan algoritma *Naive Bayes* dan *Regresi Linear* untuk mengoptimalkan efisiensi dan akurasi pengelolaan. Implementasi ini dilakukan melalui beberapa tahap.

1) Preprocessing Data

Tahap *preprocessing* data merupakan langkah awal yang krusial dalam pengolahan data untuk sistem

berbasis *Machine Learning*, guna memastikan bahwa data yang digunakan akurat, bersih, dan layak untuk dianalisis. Proses ini mencakup dua bagian utama, yaitu data *cleaning* dan data *validation*. Data *cleaning* merupakan tahap penting dalam pengolahan data karena kualitas input sangat memengaruhi akurasi hasil analisis. Konsep GIGO (*Garbage In, Garbage Out*) menegaskan bahwa data yang buruk akan menghasilkan output yang tidak akurat. Teknik data *cleaning* meliputi penghapusan entri duplikat, penanganan data hilang, perbaikan kesalahan penulisan, serta normalisasi format seperti huruf dan tanggal agar seragam [9]. Manfaat dari proses ini mencakup peningkatan akurasi analisis, mendukung pengambilan keputusan yang lebih tepat, serta efisiensi biaya dan waktu dalam pemrosesan data [9]. Sementara itu, data *validation* bertujuan untuk memastikan akurasi, konsistensi, dan keandalan data sebelum digunakan dalam sistem. Proses ini melibatkan pengecekan terhadap format yang diharapkan, keunikan entri, dan rentang nilai antar atribut untuk menjaga integritas data serta mencegah kesalahan atau duplikasi [10]. Salah satu implementasi praktis data *validation* dapat ditemukan pada fitur Validasi Data di *Microsoft Excel*, yang memungkinkan pembatasan jenis input sesuai kriteria seperti rentang angka, format tanggal, atau daftar nilai tertentu [11]. Dengan demikian, proses *preprocessing* yang mencakup data *cleaning* dan data *validation* memberikan fondasi yang valid dan seragam bagi model *machine learning*, sehingga prediksi yang dihasilkan menjadi lebih andal.

2) Implementasi *Naive Bayes*

Naive Bayes merupakan algoritma klasifikasi berbasis probabilistik yang sederhana namun efektif, yang bekerja dengan menganalisis frekuensi dan kombinasi nilai dalam dataset [12]. Algoritma ini menggunakan pendekatan statistik untuk memperkirakan kemungkinan suatu peristiwa berdasarkan data historis, sesuai dengan teori yang dikembangkan oleh Thomas Bayes [13]. *Naive Bayes* dikenal mampu menurunkan tingkat kesalahan klasifikasi dibandingkan sejumlah metode lain, dengan asumsi bahwa setiap variabel input bersifat independen terhadap yang lain [12]. Salah satu keunggulan utama dari algoritma ini adalah kebutuhan data pelatihan yang relatif rendah, karena hanya memerlukan varian dari masing-masing fitur dalam satu kelas tanpa perlu menghitung keseluruhan matriks kovarians [2]. *Naive Bayes* juga banyak digunakan dalam berbagai bidang, seperti klasifikasi dokumen teks, deteksi spam, diagnosis medis, hingga klasifikasi akademik [14][15]. Kesederhanaannya dalam implementasi menjadikannya populer di kalangan praktisi *machine learning*, meskipun terdapat beberapa keterbatasan, seperti sensitivitas terhadap probabilitas nol dan

asumsi independensi antar variabel yang tidak selalu sesuai dengan kondisi nyata [15].

Dalam konteks pengelolaan laporan di DPMPSTP Kota Surabaya, algoritma ini digunakan untuk memprediksi jenis layanan berdasarkan input seperti jenis perizinan, nama pemohon, dan dinas terkait. Implementasi algoritma *Naive Bayes* dalam penelitian ini diawali dengan proses persiapan data, yang mencakup pengumpulan data perizinan, pembersihan data dari nilai hilang dan duplikat, serta pembagian data menjadi data latih dan uji menggunakan lima skenario proporsi: 80:20, 70:30, 75:25, 85:15, dan 90:10. Model dilatih untuk mengklasifikasikan jenis layanan berdasarkan fitur-fitur yang tersedia. Perhitungan dilakukan melalui tiga tahap utama. Pertama, menghitung probabilitas a priori setiap kelas menggunakan

$$P(C) = \frac{nc}{n} \quad (1)$$

di mana $P(C)$ adalah probabilitas kelas C , nc adalah jumlah sampel dalam kelas C , dan n adalah total sampel. Kedua, menghitung probabilitas likelihood setiap fitur dengan distribusi Gaussian:

$$P(x_i | C) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(x_i - \mu_c)^2}{2\sigma_c^2}} \quad (2)$$

dengan x_i sebagai nilai fitur ke- i , μ_c sebagai rata-rata, dan σ_c sebagai standar deviasi pada kelas C . Ketiga, menghitung probabilitas posterior untuk setiap kelas dengan:

$$P(C | x) \propto P(C) \prod_{i=1}^n P(x_i | C) \quad (3)$$

dan memilih kelas dengan probabilitas tertinggi sebagai hasil klasifikasi:

$$C_{pred} = \arg \max_C P(C | x) \quad (4)$$

Model yang telah dilatih ini diintegrasikan ke dalam sistem manajemen laporan berbasis web, sehingga sistem secara otomatis memprediksi jenis layanan berdasarkan dokumen yang diinput oleh pengguna, guna mempercepat proses analisis dan pengambilan keputusan.

3) Implementasi *Regresi Linear*

Regresi Linear merupakan teknik statistik yang digunakan untuk memprediksi nilai suatu variabel dependen berdasarkan hubungan dengan satu atau lebih variabel independen [16]. Dalam metode ini, variabel independen dianggap memengaruhi variabel dependen, dan hubungan di antara keduanya diasumsikan bersifat linear [17]. Analisis *regresi linear* memodelkan keterkaitan matematis tersebut untuk menghasilkan model prediktif. Khususnya dalam konteks *regresi linear* sederhana, model hanya mempertimbangkan satu variabel independen terhadap satu variabel dependen, dan digunakan untuk memproyeksikan nilai output berdasarkan nilai input tertentu [18]. Penelitian ini menggunakan *regresi linear* sederhana karena efektivitasnya dalam

meramalkan nilai berbasis data historis dengan struktur yang relatif sederhana namun akurat. Dari sisi teoritis, penerapan metode ini diharapkan dapat memberikan kontribusi dalam pengembangan teknologi berbasis data, khususnya dalam bidang ilmu komputer.

Dari sisi praktis, *regresi linear* menjadi pendekatan yang tepat untuk merancang sistem prediktif yang efisien dan berkualitas, khususnya dalam konteks estimasi durasi pelayanan publik secara otomatis. Dalam konteks penelitian ini, algoritma *Regresi Linear* digunakan untuk memprediksi durasi penyelesaian layanan perizinan berdasarkan variabel-variabel independen seperti jenis perizinan, nama pemohon, dan waktu masuk. Tahapan implementasi dimulai dari pengumpulan data dari DPMPSTP Kota Surabaya, dilanjutkan dengan proses *preprocessing* berupa pembersihan data hilang dan duplikat, serta pembagian data menjadi data latih dan data uji dalam berbagai skenario (80:20, 70:30, 75:25, 85:15, dan 90:10) guna mengevaluasi performa model secara menyeluruh. Model *Regresi Linear* kemudian dilatih untuk memetakan hubungan antara input (fitur) dan target (durasi layanan) menggunakan persamaan umum regresi.

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n + \epsilon \quad (5)$$

di mana y adalah variabel target (durasi), x_i adalah variabel independen, β_i adalah koefisien regresi, dan ϵ adalah error. Estimasi koefisien dilakukan dengan metode kuadrat terkecil menggunakan rumus:

$$\hat{\beta} = (X^T X)^{-1} X^T y \quad (6)$$

dengan X sebagai matriks fitur, y sebagai vektor target, dan β sebagai vektor koefisien hasil pelatihan. Setelah pelatihan, model digunakan untuk memprediksi durasi layanan pada data uji, dan dievaluasi menggunakan metrik *Mean Absolute Error* (MAE) dan *Mean Absolute Percentage Error* (MAPE). Model *Regresi Linear* ini diintegrasikan ke dalam sistem manajemen laporan berbasis web, sehingga sistem dapat memprediksi durasi layanan secara otomatis berdasarkan data yang diinput oleh pengguna, guna mendukung efisiensi operasional dan pengambilan keputusan di DPMPSTP.

4) Evaluasi Hasil Pelatihan Model *Machine Learning*

Evaluasi dilakukan untuk mengukur kinerja model *Naive Bayes* dan *Regresi Linear* yang telah dibangun. Proses ini diawali dengan membagi dataset menjadi data latih dan data uji menggunakan lima skenario pembagian (80:20, 70:30, 75:25, 85:15, dan 90:10) guna memastikan kemampuan generalisasi dan menghindari overfitting. Evaluasi model klasifikasi dilakukan dengan metrik akurasi, presisi, *recall*, dan *F1-Score*. Akurasi dihitung sebagai:

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Total Jumlah Prediksi}} \quad (7)$$

Presisi dihitung dengan:

$$\text{Presisi} = \frac{\text{Jumlah Prediksi Positif Benar}}{\text{Total Jumlah Prediksi Positif}} \quad (8)$$

Recall didefinisikan sebagai:

$$\text{Recall} = \frac{\text{Jumlah Prediksi Positif Benar}}{\text{Total Jumlah Data Positif}} \quad (9)$$

F1-Score diperoleh dari rata-rata harmonik antara presisi dan *recall*:

$$\text{F1-Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (10)$$

Untuk model regresi, performa diukur menggunakan *Mean Absolute Error* (MAE) dan *Mean Absolute Percentage Error* (MAPE). MAE dihitung dengan:

$$\text{MAE}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (11)$$

sedangkan MAPE dihitung sebagai:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (12)$$

Evaluasi menyeluruh dilakukan untuk menilai akurasi, stabilitas, dan relevansi prediksi model terhadap data uji, baik dalam konteks klasifikasi maupun prediksi numerik, sehingga diperoleh pemahaman mendalam terhadap kinerja sistem yang dikembangkan.

5) Evaluasi Hasil Respon Pengguna

Setelah implementasi sistem pengelolaan laporan berbasis *Machine Learning*, evaluasi dilakukan melalui penyebaran kuesioner untuk mengukur persepsi pengguna terhadap kemudahan, kecepatan, akurasi, dan manfaat sistem. Kuesioner dirancang untuk mencakup pertanyaan seputar aspek fungsional dan pengalaman penggunaan, seperti kemudahan navigasi, ketepatan hasil klasifikasi dan prediksi, serta efisiensi waktu pemrosesan. Instrumen ini didistribusikan kepada pegawai dan staf DPMPSTP Kota Surabaya sebagai pengguna langsung sistem, khususnya mereka yang terlibat dalam pengelolaan berkas perizinan. Setelah respon dikumpulkan, dilakukan analisis data kualitatif dan kuantitatif untuk mengevaluasi sejauh mana sistem memenuhi kebutuhan pengguna serta potensi perbaikannya ke depan.

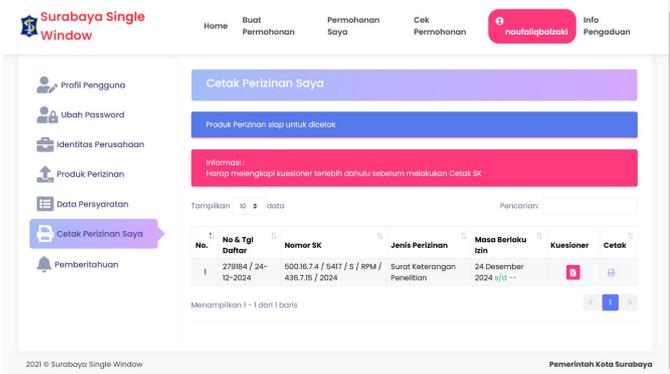
III. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil implementasi sistem pengelolaan laporan berbasis *Machine Learning* yang dikembangkan untuk meningkatkan efisiensi kerja di DPMPSTP Kota Surabaya. Sistem memanfaatkan algoritma *Naive Bayes* untuk klasifikasi jenis layanan dan *Regresi Linear* untuk prediksi estimasi durasi layanan berdasarkan data historis. Dataset yang digunakan berasal dari laporan perizinan bulan November 2024 dan telah melalui proses data *cleaning* dan validasi kualitas. Data kemudian dibagi menjadi data latih dan data uji dengan berbagai skenario pembagian. Evaluasi dilakukan menggunakan metrik akurasi, presisi, *recall*, dan *F1-score* untuk *Naive Bayes*, serta MAE dan MAPE untuk *Regresi Linear*. Hasil pengujian menunjukkan bahwa kedua model mampu memberikan prediksi yang akurat dan stabil, sehingga sistem yang dibangun efektif dalam mengotomatiskan pengolahan laporan, mengurangi proses manual, serta

mempercepat pengambilan keputusan di DPMPSTP Kota Surabaya.

A. Hasil Pengumpulan dan Preprocessing Data

Pengumpulan data dilakukan secara sistematis untuk mendukung pengembangan model *Machine Learning* dalam sistem pengelolaan laporan di DPMPSTP Kota Surabaya. Data diperoleh melalui kombinasi metode, yaitu izin resmi dari portal pelayanan publik Pemerintah Kota Surabaya (SSW Alfa) seperti pada Gbr 3.



Gbr. 3 Pendaftaran Penelitian

Studi dokumentasi terhadap arsip laporan dalam format *Excel*, serta wawancara dengan staf DPMPSTP. Data yang dikumpulkan mencakup atribut seperti nomor berkas, jenis perizinan, nama pemohon, tanggal pendaftaran, dinas terkait, dan status berkas. Wawancara dengan pegawai terkait mengungkap sejumlah kendala, seperti keterlambatan proses rekapitulasi laporan dan potensi kesalahan input data (*human error*), serta kebutuhan terhadap sistem yang lebih otomatis dan efisien. Dari keseluruhan data laporan tahun 2024, penelitian ini memfokuskan pada data bulan November dengan total 8.326 entri. Data tersebut dijadikan sebagai bahan utama pelatihan model *Naive Bayes* dan *Regresi Linear*, sebelum memasuki tahap *preprocessing* untuk memastikan kualitas dan kelengkapan data. Contoh tampilan data asli yang digunakan dalam penelitian ini dapat dilihat pada Gbr 4.

LAPORAN VERIFIKASI BERKAS MASUK PTSP SURABAYA PUSAT											
BULAN NOVEMBER TAHUN 2024											
NO.	NO. ONLINE / TANGGAL DAFTAR	NAMA PEMOHON	PERIZINAN	DINAS	KETERANGAN	BERKAS PEMOHON TERAKHIR KLIK SELESAI		PETUGAS UPTSA		NO. TELEPON PEMOHON	
						LANJUT PERLU PERBAIK	TANGGAL	JAM	TANGGAL VERIFIKASI		JAM VERIFIKASI
1	260539 / 28 November 2024 *	Lina Ayu	Izin Pemakai	Dinas Pemada	Berkas permohonan	28/11/2024	16:07:19	29/11/2024	07:36:56	LANJUT OPD	Salsa
2	260556 / 28 November 2024 *	Riski	Izin Pemakai	Dinas Pemada	Berkas permohonan	28/11/2024	16:17:07	29/11/2024	07:39:01	LANJUT OPD	Salsa
3	260564 / 28 November 2024 *	Hanifa	Izin Pemakai	Dinas Pemada	Berkas permohonan	28/11/2024	16:23:17	29/11/2024	07:39:52	LANJUT OPD	Rina
4	258375 / 25 November 2024 *	Muji Rahayu	Tera Temporal	UPTD Metrolo	Mohon arangan.	28/11/2024	20:11:34	29/11/2024	08:11:28	LANJUT OPD	Rina
5	260839 / 29 November 2024 *	Faza Budargo	Tera Ulang di	UPTD Metrolo	Berkas permohonan	29/11/2024	08:49:19	29/11/2024	08:54:16	LANJUT OPD	Rina
6	260950 / 29 November 2024 *	Riski	Tera Ulang di	UPTD Metrolo	Berkas permohonan	29/11/2024	09:45:03	29/11/2024	09:46:19	LANJUT OPD	Budi
	261043 / 29 November 2024 *	Faza	Tera Ulang di	UPTD Metrolo	Berkas permohonan					LANJUT OPD	

Gbr. 4 Data November 2024

Tahap *preprocessing* data bertujuan untuk meningkatkan kualitas dataset agar layak digunakan dalam pelatihan model. Proses ini diawali dengan data *cleaning*, yaitu penghapusan entri yang tidak valid, duplikat, atau tidak relevan. Langkah-langkahnya meliputi identifikasi dan penghapusan data

duplikat, perbaikan kesalahan penulisan, serta standarisasi format teks (huruf kapital, tanda baca, dan tanggal).

1. Penghapusan Duplikasi: Menggunakan metode `.drop_duplicates()`, seperti yang ditunjukkan pada Gbr 5, yaitu untuk menghapus entri yang sama berdasarkan nomor berkas.

```
# 1. Menghapus duplikat
data = data.drop_duplicates()
```

Gbr. 5 Hapus Data Duplikat

2. Validasi Nilai: Kolom 'NO. TELEPON PEMOHON' divalidasi agar hanya berisi karakter numerik dengan menggunakan metode `.str.isdigit()`. Baris yang tidak memenuhi kriteria ini akan dihapus dari dataset. Validasi ini memastikan bahwa hanya nomor telepon yang benar (berisi angka saja) yang diproses, sebagaimana ditunjukkan pada Gbr 6.

```
# 2. Validasi kolom NO. TELEPON PEMOHON: hanya angka
if 'NO. TELEPON PEMOHON' in data.columns:
    data['NO. TELEPON PEMOHON'] = data['NO. TELEPON PEMOHON'].astype(str)
    data = data[data['NO. TELEPON PEMOHON'].str.isdigit()]
else:
    print("Kolom 'NO. TELEPON PEMOHON' tidak ditemukan dalam dataset.")
```

Gbr. 6 Validasi Nomor Telepon Pemohon

3. Standarisasi dan Validasi Format Tanggal: Untuk memastikan konsistensi format tanggal dan memudahkan pemrosesan data lebih lanjut, setiap nilai pada kolom tanggal dikonversi menggunakan fungsi `pd.to_datetime()` dengan format 'm/d/Y'. Nilai yang tidak valid dikonversi menjadi `None` dan kemudian dihapus dari dataset. Proses ini diterapkan pada beberapa kolom, seperti yang ditunjukkan pada Gbr 7.

```
# 3. Validasi format tanggal m/d/yyyy untuk kolom tanggal
def format_valid_date(val):
    try:
        parsed = pd.to_datetime(val, errors='coerce')
        return parsed.strftime('%m/%d/%Y') if not pd.isna(parsed) else None
    except:
        return None

for date_col in ['BERKAS PEMOHON TERAKHIR KLIK SELESAI', 'PETUGAS UPTSA']:
    if date_col in data.columns:
        data[date_col] = data[date_col].apply(format_valid_date)
        data = data[data[date_col].notnull()]
    else:
        print(f"Kolom '{date_col}' tidak ditemukan.")
```

Gbr. 7 Validasi Format Tanggal

4. Standarisasi dan Validasi Format Jam: Kolom 'UNNAMED: 8' atau 'UNNAMED: 10' divalidasi agar sesuai format HH:MM:SS. Nilai tidak valid diubah menjadi `None`, lalu baris dengan nilai kosong dihapus. Jika kolom tidak ditemukan, sistem akan menampilkan

pesan peringatan. Proses ini ditunjukkan pada **Gbr 8**.

```
# 4. Validasi format jam HH:MM:SS untuk kolom jam
def format_valid_time(val):
    try:
        parsed = pd.to_datetime(val, format='%H:%M:%S', errors='coerce')
        return parsed.strftime('%H:%M:%S') if not pd.isna(parsed) else None
    except:
        return None

for time_col in ['UNNAMED: 8', 'UNNAMED: 10']:
    if time_col in data.columns:
        data[time_col] = data[time_col].apply(format_valid_time)
        data = data[data[time_col].notnull()]
    else:
        print(f"Kolom '{time_col}' tidak ditemukan.")
```

Gbr. 8 Validasi Format Jam

- Menghapus Data yang Tidak Relevan: Baris-baris dengan nilai kosong pada kolom 'KETERANGAN' dihapus menggunakan metode `.notnull()`. Langkah ini dilakukan untuk memastikan hanya data yang memiliki keterangan dipertahankan, seperti ditunjukkan pada **Gbr 9**.

```
# 5. Menghapus baris dengan KETERANGAN kosong
if 'KETERANGAN' in data.columns:
    data = data[data['KETERANGAN'].notnull()]
else:
    print("Kolom 'KETERANGAN' tidak ditemukan.")
```

Gbr. 9 Penghapusan Nilai Kosong Keterangan

- Dilakukan validasi untuk memastikan bahwa kolom nomor telepon hanya berisi karakter angka menggunakan fungsi pemeriksaan khusus, seperti ditunjukkan pada **Gbr 10**.

```
# 6. Membersihkan NAMA PEMOHON dari spasi ekstra
if 'NAMA PEMOHON' in data.columns:
    data['NAMA PEMOHON'] = data['NAMA PEMOHON'].astype(str).str.strip()
else:
    print("Kolom 'NAMA PEMOHON' tidak ditemukan.")
```

Gbr. 10 Membersihkan Spasi Ekstra Nama Pemohon

Selanjutnya, dilakukan data *validation* untuk memastikan bahwa setiap entri sesuai dengan format dan struktur yang telah ditentukan. Validasi mencakup pengecekan format data (misalnya format tanggal dan nomor telepon) serta konsistensi antar atribut dalam dataset. Setelah proses ini selesai, jumlah data valid yang siap digunakan untuk pelatihan model adalah 3.434 entri. Data hasil *preprocessing* ini dibagi menjadi data latih dan data uji dengan beberapa skenario proporsi (70:30, 80:20, 85:15, dan lainnya), guna menguji kestabilan dan akurasi dari masing-masing algoritma yang diterapkan. Rangkuman jumlah data pada setiap tahapan *preprocessing* disajikan pada **Tabel 1** berikut.

TABEL I
TOTAL DATA

Sebelum <i>Preprocessing</i>	Setelah <i>Preprocessing</i>
8.326 Data	3.434 Data

Pada **Tabel 1**, menunjukkan bahwa dari total awal 8.326 data laporan, hanya 3.434 data yang berhasil lolos proses validasi dan pembersihan, dan dinyatakan layak untuk digunakan dalam pelatihan model. Ini berarti sekitar 58,75% data dieliminasi karena tidak memenuhi standar kualitas data yang telah ditentukan. Selanjutnya, hasil akhir dari data yang telah melalui proses *preprocessing* dapat dilihat secara visual pada **Gbr 11**.

NO.	NO. ONLINE	TANGGAL DATANG	NAMA PEMOHON	PELAYAN	DURAS	KETERANGAN		BERKAS PEMOHON		PETUGAS UPTA		STATUS KETERANGAN	NO. TELEPON PEMOHON	PETUGAS	JENIS LAYANAN	DURAS
						LAJUT	PERU	TANGGAL	JAM	TANGGAL	JAM					
1	20084	10 October 2024	New	NSAFAL/USAL/240	07 Baru Pasrah Berkas Keterangan: SP7K	Dinas Kesehatan	(1) Nomor update STK Angkor (2) Nomor update silabus penyusunan silabus silabus	10/10/2024	10:19	11/10/2024	08:53	PERUBAHAN PTSP	0822072008	Sara	LAYANAN KESEHATAN	1
2	20260	21 October 2024	Normal	CHERANA/USAF/240	07 Baru Pasrah Berkas Informasi: SP7K	Dinas Kesehatan	Mohon perbaiki alamat FASIS pada saat saat pemeriksaan Surat Perijinan Sementara Pengantar Kedua pelayanan kecamatan bertamase DIBUKA kecamatan dengan petugas yang diupdate	11/10/2024	09:45	11/10/2024	09:54	PERUBAHAN PTSP	0822072008	Sara	LAYANAN KESEHATAN	1

Gbr. 11 Hasil Preprocessing Data

B. Proses Pelatihan dan Pengujian Model

Proses pelatihan dan pengujian model dilakukan untuk mengukur efektivitas algoritma *Machine Learning* yang diimplementasikan dalam sistem pengelolaan laporan DPMPTSP Kota Surabaya. Dua jenis model digunakan, yaitu *Naive Bayes* untuk klasifikasi jenis layanan dan *Regresi Linear* untuk prediksi durasi penyelesaian layanan. Dataset yang telah melalui *preprocessing* sebanyak 3.434 entri kemudian dibagi menjadi data latih dan data uji dengan beberapa skenario proporsi: 70:30, 75:25, 80:20, 85:15, dan 90:10. Pemilihan berbagai rasio pembagian ini bertujuan untuk mengevaluasi stabilitas dan generalisasi model terhadap data baru. Model *Naive Bayes* dilatih dengan menghitung probabilitas a priori, *likelihood*, dan posterior menggunakan Teorema Bayes, kemudian digunakan untuk memprediksi kelas layanan berdasarkan fitur input. Sementara itu, model *Regresi Linear* dilatih menggunakan pendekatan *ordinary least squares* untuk menghitung koefisien regresi dan digunakan untuk memprediksi durasi layanan berdasarkan kombinasi fitur numerik dan kategorikal. Setelah pelatihan, performa masing-masing model diuji menggunakan metrik evaluasi: akurasi, presisi, recall, dan *F1-score* untuk klasifikasi; serta MAE (*Mean Absolute Error*) dan MAPE (*Mean Absolute Percentage Error*) untuk prediksi. Hasil evaluasi dari proses pelatihan dan pengujian tersebut disajikan pada bagian selanjutnya.

1. Evaluasi Model *Naive Bayes*

Pengujian awal dilakukan terhadap model *Naive Bayes* tanpa penerapan teknik penyeimbangan data seperti SMOTE. Evaluasi dilakukan dengan berbagai skenario pembagian data *train-test* (*train-test split*) untuk melihat konsistensi performa model terhadap distribusi data asli. Hasil evaluasi dari setiap skenario tersebut disajikan pada **Tabel 2**.

TABEL II
SPLIT DATA NAIVE BAYES TANPA SMOTE

Test Size	Accuracy	Precision	Recall	F1-Score
80:20	0,96	0,96	0,96	0,96
70:30	0,96	0,97	0,96	0,96
75:25	0,96	0,96	0,96	0,96
85:15	0,95	0,96	0,95	0,95
90:10	0,94	0,95	0,94	0,95

Tabel 2 menyajikan evaluasi model *Naive Bayes* tanpa teknik penyeimbangan data (SMOTE)

berdasarkan lima skenario *train-test split*. Hasil menunjukkan akurasi model cukup stabil antara 0.94 hingga 0.96, dengan performa terbaik pada skenario 70:30, 75:25, dan 80:20. Presisi tertinggi, yakni 0.97, diperoleh pada skenario 70:30, mengindikasikan efektivitas model dalam menghindari klasifikasi positif palsu.

Sementara itu, nilai *Recall* stabil pada tiga skenario awal (0.96), namun menurun pada 85:15 dan 90:10, masing-masing menjadi 0.95 dan 0.94, menunjukkan penurunan kemampuan model mengenali kelas saat data latih berkurang. *F1-Score* tetap pada 0.96 di skenario awal dan menurun ke 0.95 pada skenario akhir. Secara keseluruhan, performa model menurun seiring berkurangnya proporsi data latih, dengan skenario 70:30 menjadi yang paling optimal.

Setelah pengujian awal tanpa penyeimbangan data, tahap berikutnya adalah menguji performa model *Naive Bayes* dengan menerapkan SMOTE (*Synthetic Minority Over-sampling Technique*) untuk menyeimbangkan distribusi kelas agar model tidak bias. Pengujian dilakukan pada beberapa skenario *train-test split* untuk mengevaluasi konsistensi performa model setelah penerapan SMOTE. Hasil evaluasi disajikan pada **Tabel 3**.

TABEL III
 SPLIT DATA NAIVE BAYES DENGAN SMOTE

Test Size	Accuracy	Precision	Recall	F1-Score
80:20	0,96	0,96	0,96	0,96
70:30	0,96	0,97	0,96	0,96
75:25	0,96	0,96	0,96	0,96
85:15	0,95	0,95	0,95	0,95
90:10	0,95	0,95	0,95	0,95

Tabel 3 menampilkan evaluasi model *Naive Bayes* setelah data diseimbangkan menggunakan SMOTE, diuji dalam lima skenario *train-test split*. Hasil menunjukkan akurasi tetap tinggi dan stabil, mencapai 0.96 pada skenario 70:30, 75:25, dan 80:20, serta sedikit menurun menjadi 0.95 pada skenario 85:15 dan 90:10. Presisi juga konsisten tinggi, dengan nilai tertinggi 0.97 pada skenario 70:30, menandakan ketepatan prediksi tetap baik meskipun data sintetis ditambahkan.

Recall mengalami peningkatan dibandingkan skenario tanpa SMOTE, stabil di kisaran 0.96–0.95 di semua skenario. Hal ini mencerminkan kemampuan model yang lebih baik dalam mengenali seluruh kelas setelah distribusi data diseimbangkan. *F1-Score* juga menunjukkan performa yang baik, yakni 0.96 pada tiga skenario awal dan 0.95 pada dua skenario terakhir. Secara keseluruhan, penerapan SMOTE berhasil meningkatkan sensitivitas terhadap kelas minoritas tanpa mengorbankan keseimbangan kinerja model.

Secara umum, penerapan SMOTE berhasil meningkatkan generalisasi model terhadap data yang tidak seimbang, tanpa menyebabkan penurunan signifikan pada akurasi atau metrik lainnya. Skenario 70:30 tetap menjadi skenario terbaik karena menghasilkan nilai Presisi dan *F1-Score* tertinggi.

Hasil pengujian menunjukkan bahwa model *Naive Bayes* memiliki performa yang stabil di seluruh skenario pembagian data, dengan akurasi tinggi berkisar antara 95% hingga 96%. Presisi tertinggi sebesar 0.97 tercapai pada skenario 70:30, baik tanpa maupun dengan SMOTE, menunjukkan kemampuan model yang sangat baik dalam meminimalkan kesalahan klasifikasi positif palsu. Nilai *Recall* dan *F1-Score* juga stabil pada skenario 70:30, 75:25, dan 80:20, serta mengalami sedikit penurunan saat proporsi data latih semakin kecil. Penerapan SMOTE membantu menjaga kestabilan performa, khususnya dalam mengenali kelas minoritas. Dari seluruh skenario, 70:30 merupakan konfigurasi terbaik karena memberikan nilai tertinggi dan paling konsisten pada semua metrik evaluasi.

2. Evaluasi Model *Regresi Linear*

Model *Regresi Linear* diuji untuk melihat seberapa baik modelnya dapat memprediksi durasi waktu berdasarkan data historis. Berikut merupakan hasil evaluasi model menggunakan beberapa skenario pembagian data *train-test split* yang disajikan pada **Tabel 4**.

TABEL IV
 SPLIT DATA REGRESI LINEAR

Test Size	MAE Train	Test Size	MAE Train	Test Size
80:20	0,15	7,28%	0,33	9,51%
70:30	0,15	7,30%	0,28	9,26%
75:25	0,16	7,55%	0,29	10,43%
85:15	0,15	7,48%	0,38	8,42%
90:10	0,22	10,85%	0,24	18,17%

Tabel 4 menunjukkan hasil evaluasi model *Regresi Linear* berdasarkan lima skenario pembagian data *train-test*, dengan metrik MAE (*Mean Absolute Error*) dan MAPE (*Mean Absolute Percentage Error*) baik untuk data pelatihan (train) maupun pengujian (test). Hasil menunjukkan bahwa MAE dan MAPE pada data pelatihan cenderung rendah dan stabil di kisaran 0.15 hari dan 7% hingga 7.5%, kecuali pada skenario 90:10 yang menunjukkan lonjakan MAE (0.22 hari) dan MAPE (10.85%). Ini menunjukkan bahwa model memiliki kemampuan prediksi yang cukup akurat selama pelatihan, tetapi performanya memburuk saat data latih terlalu sedikit.

Pada data pengujian, MAE dan MAPE cenderung bervariasi lebih besar. Nilai terbaik untuk MAE test (0.24 hari) dan MAPE test (8.42%) diperoleh pada skenario 90:10 dan 85:15. Namun, meskipun MAE test 90:10 lebih rendah, nilai MAPE-nya justru

tertinggi (18.17%), menunjukkan bahwa kesalahan relatif model terhadap nilai aktual cukup besar. Sebaliknya, skenario 70:30 menghasilkan keseimbangan yang baik, dengan MAE *test* terendah (0.28 hari) dan MAPE *test* kedua terendah (9.26%). Hal ini mengindikasikan bahwa skenario 70:30 memberikan hasil paling stabil dan seimbang dalam hal akurasi dan kesalahan relatif, menjadikannya konfigurasi optimal untuk model *Regresi Linear* dalam konteks ini. Berdasarkan interpretasi MAPE menurut Lewis dalam artikel oleh Ajiono dan Hariguna pada tahun 2023, nilai MAPE dapat dikategorikan ke dalam beberapa indikator, yaitu <10% (Sangat Akurat), 10–20% (Baik), 20–50% (Rata-rata), dan >50% (Tidak Akurat), sebagaimana ditampilkan pada **Tabel 5** [19].

TABEL V
 KRITERIA PENILAIAN MODEL NILAI MAPE

MAPE	Indikator
<10%	Sangat Akurat
10% - 20%	Baik
20% - 50%	Rata-Rata
>50%	Tidak Akurat

3. Kesimpulan Hasil Pengujian Model *Machine Learning*

Berdasarkan hasil evaluasi, model *Naive Bayes* menunjukkan performa klasifikasi yang sangat baik dengan akurasi stabil antara 95%–96% di berbagai skenario. Nilai presisi tertinggi sebesar 97% dicapai pada skenario pembagian data 70:30, yang mencerminkan kemampuannya dalam mengklasifikasikan jenis layanan secara akurat dan konsisten. Model ini juga menunjukkan nilai *recall* tertinggi sebesar 96% pada skenario 80:20, 70:30, dan 75:25, menandakan sensitivitas yang tinggi dalam mendeteksi kasus positif, serta *F1-Score* yang stabil pada angka 96%, menunjukkan keseimbangan yang baik antara presisi dan *recall*. Sementara itu, model *Regresi Linear* juga menunjukkan performa terbaik pada skenario 70:30, dengan *Mean Absolute Error* (MAE) berkisar antara 0,25 hingga 0,30 hari dan *Mean Absolute Percentage Error* (MAPE) sebesar 8,42%, yang menunjukkan kemampuan prediksi durasi layanan secara cukup akurat. Berdasarkan hasil tersebut, kedua model dengan skenario 70:30 ditetapkan sebagai model terbaik dan digunakan dalam implementasi sistem pengelolaan laporan di DPMPSTP Kota Surabaya, untuk meningkatkan efisiensi, akurasi, dan kecepatan dalam proses pelayanan publik yang berbasis data.

C. Implementasi Model ke dalam Sistem

Setelah proses pelatihan dan evaluasi selesai, model *Naive Bayes* dan *Regresi Linear* diimplementasikan ke dalam sistem pengelolaan laporan berbasis web yang dirancang khusus untuk kebutuhan DPMPSTP Kota Surabaya. Model yang telah dilatih

disimpan dalam format .pkl dan dikonversi menjadi .json untuk integrasi ke backend sistem berbasis *Laravel*. Proses integrasi dilakukan menggunakan pendekatan modular, di mana *Python* digunakan sebagai pemroses utama untuk eksekusi model, sedangkan *React.js* digunakan pada sisi antarmuka pengguna untuk menampilkan hasil klasifikasi dan prediksi secara *real-time*. Sistem memungkinkan pengguna mengunggah file laporan dalam format *Excel* atau memasukkan data secara manual melalui form. Setelah data diunggah, sistem secara otomatis akan mengklasifikasikan jenis layanan menggunakan model *Naive Bayes* dan memprediksi estimasi durasi layanan menggunakan *Regresi Linear*. Hasil prediksi ditampilkan langsung kepada pengguna melalui dashboard yang interaktif, bersama informasi pendukung seperti nama pemohon, jenis izin, dan status berkas. Implementasi ini mempercepat proses validasi dan pengolahan laporan, mengurangi beban kerja manual, serta meningkatkan transparansi dan akurasi pengambilan keputusan dalam pengelolaan dokumen perizinan.

1) Arsitektur Sistem

Sistem pengelolaan laporan berbasis *Machine Learning* yang dikembangkan dalam penelitian ini dirancang sebagai aplikasi web dengan *backend* menggunakan *Laravel* dan model *Machine Learning* terintegrasi dalam bahasa *Python*, serta antarmuka pengguna yang dibangun dengan *React.js*. Arsitektur sistem terdiri dari lima tahapan utama. Pertama, tahap persiapan *database* (*MySQL*) dilakukan dengan menambahkan dua kolom utama, yaitu *ml service type* untuk menyimpan hasil prediksi jenis layanan dari model *Naive Bayes* dan *ml duration* untuk menyimpan hasil estimasi durasi layanan dari model *Regresi Linear*. Kedua, pada tahap input data oleh pengguna, pengguna berinteraksi melalui antarmuka *React.js* untuk menampilkan hasil prediksi secara langsung dalam bentuk pop-up pada setiap baris data. Hasil klasifikasi oleh *Naive Bayes* ditampilkan seperti pada **Gbr 12**,

```
<TableRow className="flex items-center gap-2 p-2 dark:bg-neutral-900">
  <p className="font-medium min-w-32">Jenis Layanan</p>
  <p>
    <span className="mr-2"></span>
    {data.ml_service_type || data.service_type || 'Tidak tersedia'}
  </p>
</TableRow>
```

Gbr. 12 Menampilkan Prediksi Naive Bayes

sedangkan hasil prediksi durasi oleh *Regresi Linear* ditunjukkan pada **Gbr 13**.

```
<TableRow className="flex items-center gap-2 p-2 hover:bg-inherit">
  <p className="font-medium min-w-32">Durasi</p>
  <p>
    <span className="mr-2"></span>
    {data.ml_duration || data.duration || 'Tidak tersedia'}
  </p>
</TableRow>
```

Gbr. 13 Menampilkan Prediksi Regresi Linear

Ketiga, pada tahap pengolahan dan prediksi, sistem menerima input dari *React.js* dan meneruskannya ke skrip *Python* menggunakan fungsi `shell_exec()`. Jika input berupa file *Excel*, sistem mengirimkan path file tersebut untuk diproses seperti pada **Gbr 14**,

```
// Execute the Python script with the file path as an argument
$pythonScript = base_path(path: 'process_excel.py'); // Adjust the path if needed
$command = escapeshellcmd(command: "python \"$pythonScript\" \"$filePath\"");
$sheetData = shell_exec(command: $command . " 2>&1");
Log::info(message: "Python script output: " . $sheetData);

// Capture both stdout and stderr
if (is_string(value: $sheetData)) {
    Log::info(message: "Spreadsheet is a string, attempting to decode...");
    $sheetData = json_decode(json: $sheetData, associative: true);
}
```

Gbr. 14 Implementasi Fitur Import

sedangkan untuk input manual, data dikirim satu per satu untuk diproses secara instan seperti pada **Gbr 15**.

```
$pythonScript = base_path(path: 'process_excel.py'); // Adjust the path if needed
$command = escapeshellcmd(command: "python \"$pythonScript\" \"$None\" \"$izin\" \"$nama");
$output = shell_exec(command: $command . " 2>&1");
Log::info(message: "Python script output: " . $output);

// Capture both stdout and stderr
if (is_string(value: $output)) {
    Log::info(message: "Output is a string, attempting to decode...");
    $output = json_decode(json: $output, associative: true);
}

// Ambil hasil ML
$naiveBayesResult = $output[0] ?? 'Tidak tersedia';
$linearRegressionResult = $output[1] ?? 'Tidak tersedia';

Log::info(message: "Naive Bayes Prediction (Jenis Layanan): " . $naiveBayesResult);
Log::info(message: "Linear Regression Prediction (Durasi): " . $linearRegressionResult);

// Tambahkan hasil ML ke dalam data yang akan disimpan
$valid['ml_service_type'] = $naiveBayesResult;
$valid['ml_duration'] = $linearRegressionResult;
```

Gbr. 15 Implementasi Fitur Manual

Keempat, tahap pemrosesan *Machine Learning* dilakukan dengan memanggil model yang telah dilatih sebelumnya dan disimpan dalam format `.pkl` dan `.json`, sehingga tidak perlu dilatih ulang setiap kali dijalankan. Model *Naive Bayes* memprediksi jenis layanan berdasarkan fitur seperti jenis perizinan, dinas, dan nama pemohon seperti pada **Gbr 16**,

```
def naiveBayes(dft) :
    #Siapkan kamus jenis layanan
    with open('nb_dict.json') as f:
        nb_dict = json.load(f)

    # Encode fitur (X)
    dft['PERIZINAN'] = encodeThis(dft['PERIZINAN'], nb_dict['PERIZINAN'])
    dft['DINAS'] = encodeThis(dft['DINAS'], nb_dict['DINAS'])
    dft['NAMA PEMOHON'] = encodeThis(dft['NAMA PEMOHON'], nb_dict['NAMA PEMOHON'])

    # Definiskan fitur
    X = dft[['PERIZINAN', 'DINAS', 'NAMA PEMOHON']]

    #Muat model dan prediksi
    model = joblib.load("model_naive_bayes.pkl")
    prediction = model.predict(X)

    prediction_encoded = []
    for i in prediction :
        try :
            prediction_encoded.append(nb_dict["JENIS LAYANAN"][str(i)])
        except :
            prediction_encoded.append("Layanan")

    # prediction_encoded = [nb_dict["JENIS LAYANAN"][str(i)] for i in prediction]

    result = prediction_encoded

    return result
```

Gbr. 16 Baris kode Naive Bayes Pada File Python

sedangkan *Regresi Linear* memperkirakan durasi layanan berdasarkan kombinasi fitur teks, tanggal, dan waktu seperti pada **Gbr 17**.

```
def linearReg(dft) :
    # Pilih fitur teks dan kolom tanggal
    text_columns = ["PERIZINAN", "DINAS", "KETERANGAN", "JENIS LAYANAN"]
    date_columns = ["BERKAS PEMOHON TERAKHIR KLIK SELESAI", "PETUGAS UPTISA"]
    time_columns = ["Unnamed: 0", "Unnamed: 10"]

    # Konversi tanggal dengan format n/d/yyyy
    dft["BERKAS PEMOHON TERAKHIR KLIK SELESAI"] = pd.to_datetime(dft["BERKAS PEMOHON TERAKHIR KLIK SELESAI"], format="%m/%d/%Y", errors='coerce')
    dft["PETUGAS UPTISA"] = pd.to_datetime(dft["PETUGAS UPTISA"], format="%m/%d/%Y", errors='coerce')

    # Tambahkan fitur hari dalam seminggu
    dft["HARI_MASUK"] = dft["BERKAS PEMOHON TERAKHIR KLIK SELESAI"].dt.dayofweek
    dft["HARI_VERIF"] = dft["PETUGAS UPTISA"].dt.dayofweek

    # Konversi jam masuk dan jam verifikasi ke format datetime.time
    dft["JAM_MASUK"] = pd.to_datetime(dft["Unnamed: 0"], format="%H:%M:%S", errors='coerce').dt.time
    dft["JAM_VERIF"] = pd.to_datetime(dft["Unnamed: 10"], format="%H:%M:%S", errors='coerce').dt.time

    # Hitung DURASI sebagai target variabel
    dft["DURASI"] = (dft["PETUGAS UPTISA"] - dft["BERKAS PEMOHON TERAKHIR KLIK SELESAI"]).dt.days

    # Hitung selisih waktu dalam jam antara jam masuk dan verifikasi
    dft["SELISIH_JAM"] = dft.apply(calculate_time_difference, axis=1)

    # Konversi jam ke format numerik dalam jam desimal
    dft["JAM_MASUK_DECIMAL"] = dft["JAM_MASUK"].astype(str).apply(lambda x: 1 if 9 <= x <= 16 else 0)
    dft["JAM_VERIF_DECIMAL"] = dft["JAM_VERIF"].astype(str).apply(lambda x: 1 if 9 <= x <= 16 else 0)

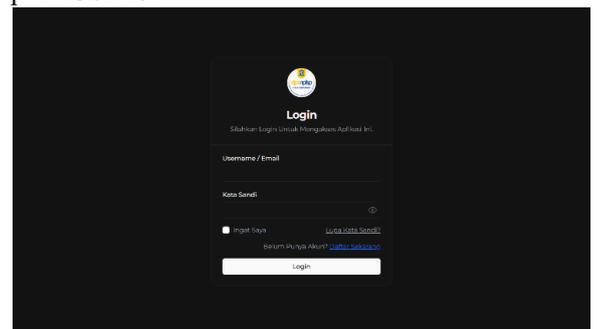
    # Tambahkan indikator waktu sibuk (jam kerja 09:00 - 16:00)
    dft["WAKTU_SIBUK_MASUK"] = dft["JAM_MASUK_DECIMAL"].apply(lambda x: 1 if 9 <= x <= 16 else 0)
    dft["WAKTU_SIBUK_VERIF"] = dft["JAM_VERIF_DECIMAL"].apply(lambda x: 1 if 9 <= x <= 16 else 0)
```

Gbr. 17 Baris Kode Regresi Linear Pada File Python

Terakhir, pada tahap penyimpanan dan penampilan hasil, sistem menyimpan hasil prediksi yang diterima dari *Python* ke dalam *database* melalui kolom `ml service type` dan `ml duration`. Selanjutnya, data hasil prediksi dikembalikan ke antarmuka *React.js* dan ditampilkan secara otomatis kepada pengguna sebagai bagian dari proses kerja sistem yang interaktif dan *real-time*.

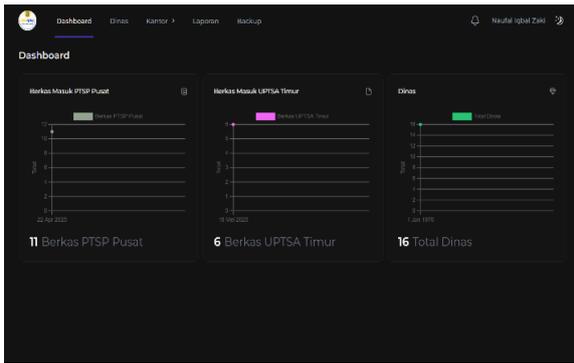
2) Alur Pemrosesan Data hingga Prediksi di Website

Sistem website pengelolaan laporan ini dirancang dengan antarmuka yang ramah pengguna dan alur kerja yang sistematis untuk memudahkan proses input data hingga memperoleh hasil prediksi. Proses dimulai dengan tahapan login atau registrasi akun pengguna. Bagi pengguna baru, registrasi dapat dilakukan melalui formulir yang tersedia dan dilanjutkan dengan verifikasi akun yang bisa dilihat pada **Gbr 18**.



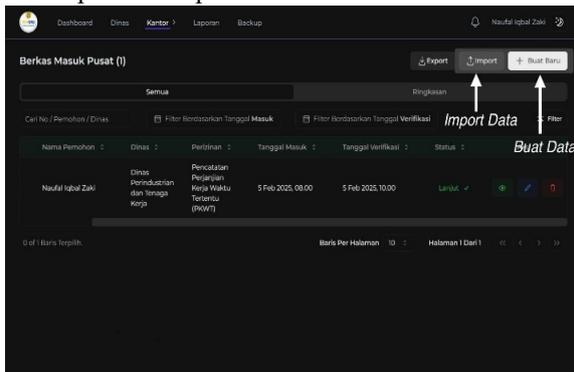
Gbr. 18 Halaman Login

Setelah berhasil masuk, pengguna diarahkan ke halaman dashboard, yang menampilkan ringkasan data laporan dalam bentuk grafik visualisasi guna mempermudah pemantauan kondisi layanan secara umum pada **Gbr 19**.



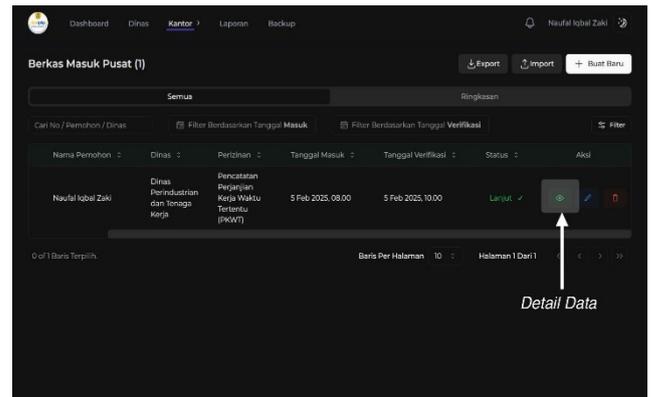
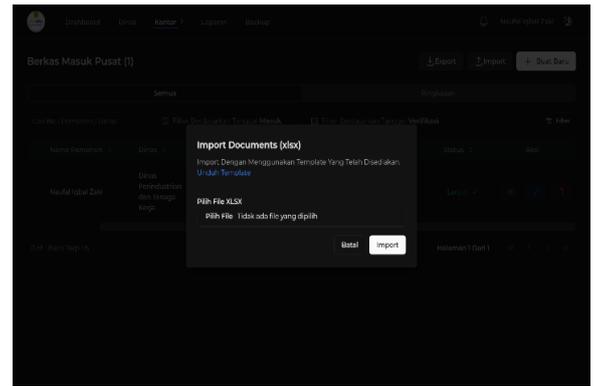
Gbr. 19 Halaman Dashboard

Untuk mengelola laporan, pengguna dapat mengakses menu “Kantor” **Gbr 4.20**, yang menyediakan dua opsi utama: import data dan tambah data. Fitur import data memungkinkan pengguna mengunggah file *Excel (.xlsx)* berisi kumpulan data laporan sekaligus, dengan format yang telah disesuaikan menggunakan template yang dapat diunduh langsung. Sementara itu, fitur tambah data ditujukan bagi pengguna yang ingin menambahkan entri laporan satu per satu melalui formulir manual.



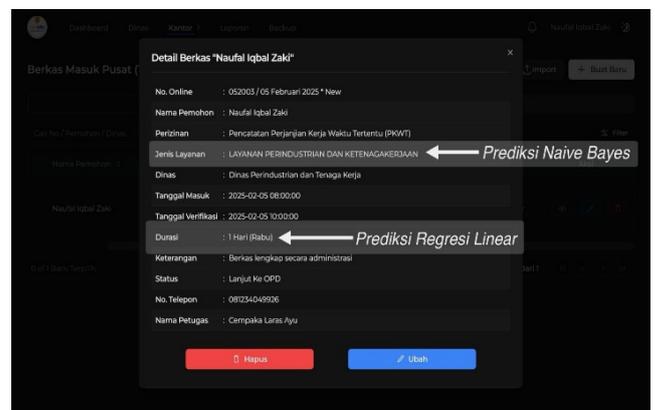
Gbr. 20 Halaman Kantor

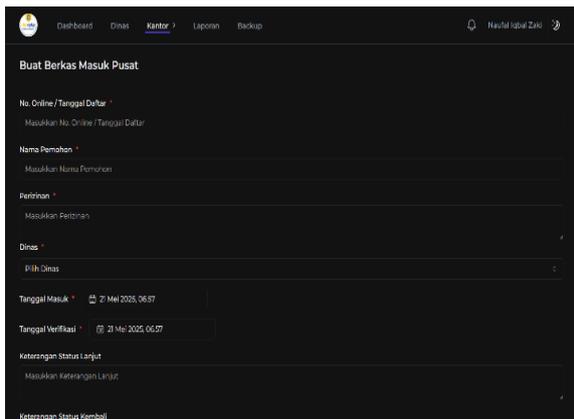
Setelah file diunggah atau data ditambahkan secara manual, sistem secara otomatis membaca, menyimpan ke *database*, dan memproses data menggunakan model *Machine Learning*. Proses ini berlangsung cepat tanpa intervensi tambahan dari pengguna yang ditunjukkan pada **Gbr 21** yaitu menunjukkan import



Gbr. 21 Import Data

data dan **Gbr 22** menunjukkan input data manual.





Gbr. 22 Halaman Tambah Data

Seluruh data laporan yang telah masuk ditampilkan dalam tabel pada halaman “Kantor”. Untuk melihat hasil prediksi, pengguna cukup mengklik ikon “eye” di setiap baris laporan seperti **Gbr 23**.

Gbr. 23 Icon Eye Untuk Detail

Detail laporan kemudian ditampilkan dalam tampilan *popup* yang mencakup hasil klasifikasi jenis layanan menggunakan model *Naive Bayes*, dan estimasi durasi layanan dari model *Regresi Linear* yang bisa dilihat pada **Gbr 24**.

Gbr. 24 Hasil Prediksi Machine Learning

Dengan alur kerja yang ringkas dan otomatis, sistem ini memungkinkan pengguna untuk mengunggah atau memasukkan data secara efisien serta langsung mendapatkan hasil prediksi yang informatif. Hal ini tidak hanya mempercepat proses pengelolaan laporan, tetapi juga meningkatkan akurasi data serta mendukung pengambilan keputusan secara cepat dan berbasis data.

D. Evaluasi Kinerja Sistem (Uji Kuisisioner Pengguna)

Evaluasi kinerja sistem dilakukan dengan metode kuisisioner untuk mengetahui sejauh mana sistem pengelolaan laporan berbasis *Machine Learning* yang telah diimplementasikan dapat memenuhi kebutuhan operasional pengguna di DPMPTSP Kota Surabaya. Kuisisioner disebarkan kepada 18 orang responden yang terdiri dari pegawai atau staf yang terlibat langsung dalam pengelolaan laporan perizinan. Instrumen evaluasi mencakup beberapa aspek, yaitu: kemudahan penggunaan sistem, akurasi hasil klasifikasi dan prediksi, kecepatan pemrosesan, tampilan antarmuka, serta manfaat sistem secara umum terhadap alur kerja.

Hasil kuisisioner menunjukkan bahwa seluruh responden (100%) menyatakan sistem ini mudah digunakan, dan 94,6% responden menyatakan bahwa sistem mampu mengurangi kesalahan manual dalam pengolahan laporan. Selain itu, seluruh responden juga menyatakan bahwa sistem membantu mempercepat proses rekapitulasi bulanan, serta meningkatkan akurasi prediksi layanan. Rangkuman hasil persentase jawaban

pengguna terhadap indikator kuisisioner ditampilkan pada **Tabel 6**.

TABEL VI
REKAPITULASI HASIL KUISISIONER

No	Pertanyaan	Respon Mayoritas
1	Sistem mengurangi kesalahan manual	61,1% Setuju, 33,3% Sangat Setuju, 5,6% Netral
2	Data valid dan akurat	55,6% Setuju, 38,9% Sangat Setuju, 5,6% Netral
3	Sistem mudah digunakan	55,6% Sangat Setuju, 44,4% Setuju
4	Sistem membantu pengelolaan laporan	55,6% Setuju, 44,4% Sangat Setuju
5	Waktu pemrosesan menjadi lebih cepat	61,1% Setuju, 38,9% Sangat Setuju
6	Sistem stabil tanpa kendala teknis	50% Setuju, 38,9% Sangat Setuju, 11,1% Netral
7	Kualitas prediksi jenis layanan	50% Baik, 50% Sangat Baik
8	Kepercayaan terhadap hasil prediksi durasi	61,1% Percaya, 38,9% Sangat Percaya
9	Kepuasan terhadap sistem secara keseluruhan	61,1% Sangat Puas, 38,9% Puas

Temuan ini menunjukkan bahwa sistem tidak hanya valid secara teknis, tetapi juga diterima dengan baik secara praktis oleh pengguna. Analisis evaluatif terhadap persepsi pengguna berdasarkan hasil kuisisioner dapat dilihat pada **Tabel 7**, yang menunjukkan tingkat kepuasan pengguna terhadap setiap aspek fungsional sistem serta potensi pengembangan ke depan. Secara keseluruhan, sistem berkontribusi nyata dalam meningkatkan efisiensi kerja dan mendukung digitalisasi pelayanan publik di lingkungan DPMPTSP Kota Surabaya.

TABEL VII
HASIL ANALISIS KUISISIONER

No	Pertanyaan	Mean	Std. Dev
P1	Sistem mengurangi kesalahan manual	4,28	0,56
P2	Data dalam sistem valid dan akurat	4,33	0,58
P3	Sistem mudah digunakan	4,56	0,50
P4	Sistem membantu dalam pengelolaan laporan	4,44	0,50
P5	Proses pemrosesan lebih cepat	4,39	0,49
P6	Sistem bekerja secara stabil tanpa kendala teknis	4,28	0,65
P7	Prediksi jenis layanan akurat	4,50	0,50
P8	Prediksi waktu penyelesaian sesuai harapan	4,39	0,49

P9	Kepuasan umum terhadap sistem	4,61	0,49
----	-------------------------------	------	------

Berdasarkan hasil analisis, seluruh indikator memiliki nilai rata-rata di atas 4,20, menunjukkan bahwa responden cenderung memberikan penilaian positif terhadap sistem. Indikator dengan rata-rata tertinggi adalah kepuasan umum terhadap sistem (P9) sebesar 4,61, yang mencerminkan tingkat penerimaan yang sangat baik. Sementara itu, indikator stabilitas sistem (P6) memiliki simpangan baku tertinggi (0,65), menandakan adanya sedikit perbedaan pengalaman teknis antar pengguna. Secara keseluruhan, sistem *Machine Learning* dinilai efektif, mudah digunakan, dan bermanfaat dalam mendukung efisiensi pengelolaan laporan.

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai implementasi sistem pengelolaan laporan berbasis *Machine Learning* pada website laporan di Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu (DPMPTSP) Kota Surabaya, maka diperoleh beberapa kesimpulan sebagai berikut:

1. Proses implementasi model *Machine Learning* telah berhasil dilakukan dengan menggunakan algoritma *Naive Bayes* untuk klasifikasi jenis layanan dan *Regresi Linear* untuk prediksi durasi waktu. Model yang digunakan merupakan hasil terbaik dari evaluasi dengan skenario pembagian data 70:30, yang menunjukkan kinerja paling optimal dalam hal akurasi dan ketepatan prediksi. Tahapan implementasi mencakup pengumpulan data, *preprocessing*, pelatihan model, dan integrasi ke dalam sistem berbasis web, yang berjalan secara efektif dan sesuai dengan kebutuhan operasional di DPMPTSP Kota Surabaya.
2. Hasil pengujian menunjukkan performa yang baik pada kedua model. Algoritma *Naive Bayes* mampu mencapai akurasi hingga 96%, dengan presisi tertinggi sebesar 97% serta *recall* yang tinggi pada skenario 70:30 dan 80:20, menunjukkan kemampuannya dalam mengklasifikasikan data secara akurat dan seimbang. Sementara itu, model *Regresi Linear* juga menunjukkan kinerja unggul pada skenario 70:30, dengan nilai MAE sebesar 0,28 hari dan MAPE hanya 9,26%, yang termasuk dalam kategori sangat baik untuk prediksi durasi layanan. Dengan demikian, dapat disimpulkan bahwa skenario 70:30 memberikan hasil paling optimal untuk kedua algoritma, baik dalam konteks klasifikasi maupun prediksi, karena mampu menjaga keseimbangan antara akurasi, presisi, dan kesalahan prediksi secara keseluruhan.
3. Tanggapan pengguna terhadap sistem sangat positif. Berdasarkan hasil kuesioner, sebagian besar responden menyatakan bahwa sistem ini mudah digunakan, mempercepat proses pengelolaan laporan, dan memberikan hasil prediksi yang akurat. Seluruh

responden merasa puas terhadap sistem, serta memberikan saran yang membangun untuk pengembangan lebih lanjut.

V. SARAN

Untuk meningkatkan efektivitas sistem pengelolaan laporan berbasis *Machine Learning* dan memaksimalkan manfaat yang dihasilkan, berikut beberapa saran yang dapat diterapkan:

1. Peningkatan Kualitas Data dan Automasi Validasi
Untuk mengatasi tantangan terkait kualitas data, diperlukan pengembangan sistem data *cleaning* otomatis yang dapat mendeteksi duplikasi, mengisi nilai yang hilang, serta memastikan format data yang seragam sebelum diproses oleh model *Machine Learning*. Selain itu, sistem validasi otomatis berbasis aturan dapat dikembangkan untuk mengurangi ketergantungan pada validasi manual oleh staf.
2. Peningkatan Infrastruktur Teknologi
Implementasi sistem berbasis *Machine Learning* memerlukan infrastruktur yang andal. Oleh karena itu, diperlukan penguatan kapasitas server, peningkatan kecepatan pemrosesan, serta optimasi penyimpanan data agar sistem dapat berjalan dengan lebih stabil dan efisien.
3. Evaluasi Berkelanjutan dan Pengembangan Sistem
Untuk memastikan sistem tetap optimal, diperlukan evaluasi berkala terhadap performa model dan tingkat kepuasan pengguna. Selain itu, eksplorasi model *Machine Learning* lainnya seperti *Random Forest*, *Support Vector Machine* (SVM), atau *Gradient Boosting* dapat dipertimbangkan untuk meningkatkan akurasi klasifikasi dan prediksi. Di sisi lain, penerapan *Natural Language Processing* (NLP) juga dapat membantu dalam pemrosesan teks, seperti ekstraksi informasi otomatis dan pemahaman konteks dokumen. Dengan evaluasi dan pengembangan berkelanjutan, sistem pengelolaan laporan dapat semakin efektif dalam mendukung kebutuhan DPMPTSP Kota Surabaya.

Dengan penerapan saran-saran ini, diharapkan sistem pengelolaan laporan berbasis *Machine Learning* dapat lebih optimal dalam mendukung efisiensi dan efektivitas pelayanan publik, khususnya dalam pengelolaan laporan perizinan di DPMPTSP Kota Surabaya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Dinas Penanaman Modal dan Pelayanan Terpadu Satu Pintu (DPMPTSP) Kota Surabaya yang telah memberikan izin dan dukungan penuh dalam pengumpulan data serta pelaksanaan penelitian ini. Terima kasih juga disampaikan kepada dosen pembimbing yang telah memberikan arahan, masukan, dan motivasi selama proses penelitian berlangsung. Penulis turut mengapresiasi partisipasi para staf dan pegawai DPMPTSP yang bersedia menjadi responden dalam evaluasi sistem. Akhir kata, penghargaan diberikan kepada Tim JIEET yang telah

menyediakan template jurnal ini dan mempermudah penulis dalam proses penulisan artikel ilmiah.

REFERENSI

- [1] DPM-PTSP Surabaya. (n.d.). <https://dpm-ptsp.surabaya.go.id/>.
- [2] D. A. Pratiwi, R. M. Awangga, and M. Y. H. Setyawan, Seleksi Calon Kelulusan Tepat Waktu Mahasiswa Teknik Informatika Menggunakan Naive Bayes, Pertama. Bandung:Kreatif Industri Nusantara, 2020.
- [3] Saputra, S. A., & Soetanto, H. (2024). Implementasi Naïve Bayes untuk Klasifikasi Prediksi serta Analisis Data Banjir di Wilayah Jakarta Pusat. 5th Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI), 3(2), 296–304. <https://senafiti.budiluhur.ac.id/index.php/senafiti/index>.
- [4] Arief Nurdini, & Anita. (2022). Analisis Peramalan Permintaan Tempe Gmo 450 Gram Dengan Menggunakan Metode Regresi Linear. Jurnal Ilmiah Teknik, 1(2), 131–142. <https://doi.org/10.56127/juit.v1i2.203>.
- [5] Fawwazrin, A. M. (2024). Penerapan Metode Regresi Linier untuk Prediksi Waktu Pencetakan KTP Disdukcapil Kecamatan Samarinda Utara. Skripsi Sarjana, Program Studi Teknik Informatika, Universitas Muhammadiyah Kalimantan Timur.
- [6] Telaumbanua, F. D., Hulu, P., Nadeak, T. Z., Lumbantong, R. R., & Dharma, A. (2019). Penggunaan Machine Learning Di Bidang Kesehatan. Jurnal Teknologi Dan Ilmu Komputer Prima (JUTIKOMP), 2(2), 391-399.
- [7] Retnoningsih, E., & Pramudita, R. (2020). Mengenal Machine Learning Dengan Teknik Supervised dan Unsupervised Learning Menggunakan Python. Bina Insani Ict Journal, 7(2), 156-165.
- [8] Ahmad, A. (2017). Mengenal artificial intelligence, machine learning, neural network, dan deep learning. J. Teknol. Indones., no. October, 3.
- [9] Risyad, S. A. (2023, June 3). Data Cleaning: Arti, Manfaat, dan Cara Melakukannya. Data Cleaning: Arti, Manfaat, dan Cara Melakukannya. <https://dibimbing.id/blog/detail/data-cleaning-arti-manfaat-dan-cara-melakukannya>.
- [10] Apa itu Validasi Data ? Arti, Fungsi, Contoh, FAQs 2025 | RevouU. (n.d.-a). RevouU. <https://www.revou.co/kosakata/validasi-data>.
- [11] Sofalvi, S., & Schueler, H. E. (2021). Assessment of Bioanalytical Method Validation Data Utilizing Heteroscedastic Seven-Point Linear Calibration Curves by EZSTATSG1 Customized Microsoft Excel Template. Journal of Analytical Toxicology, 45(8), 772–779.
- [12] Titimeidara, M. Y., & Hadikurniawati, W. (2021). Implementasi Metode Naive Bayes Implementasi Metode Naive Bayes Classifier Untuk Klasifikasi Status Gizi Stunting Pada Balita. Jurnal Ilmiah Informatika , 50241(1).
- [13] Cholid Wahyudin, W. (N.D.). Klasifikasi Stunting Balita Menggunakan Naive Bayes dengan Seleksi Fitur Forward Selection. In Jurnal Bisnis Digital Dan Sistem Informasi (Vol. 1).
- [14] R. Rachman and R. N. Handayani, "Klasifikasi algoritma Naive Bayes dalam memprediksi tingkat kelancaran pembayaran sewa teras UMKM," Jurnal Informatika, vol. 8, no. 2, pp. 111–122, Sep. 2021.
- [15] Alfa Saleh. Implementasi Metode Klasifikasi Naive Bayes dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga. Citec Journal, 2010, 2 (3), 209.
- [16] A. Muqtadir, A. A. Suryanto. 2018, Penerapan Regresi Linear Untuk Memprediksi Kebutuhan Produksi Padi, Vol. 3, hal. 331–332.
- [17] N. Kusumawati, F. Marisa, dan I. D. Wijaya, 2017, Prediksi Kurs Rupiah Terhadap Dollar Amerika Dengan Menggunakan Metode Regresi Linear, J I M P - J. Inform. Merdeka Pasuruan, Vol. 2, No. 3, hal. 45–56, doi: 10.37438/jimp.v2i3.79.
- [18] Ayuni, G. N., & Fitriyah, D. (2019). Penerapan metode Regresi Linear untuk prediksi penjualan properti pada PT XYZ. Jurnal Telematika, 14(2), 79–86. <https://journal.itb.ac.id/telematika/article/view/321>.
- [19] Ajiono, A., & Hariguna, T. (2023). Comparison of Three Time Series Forecasting Methods on Linear Regression, Exponential Smoothing and Weighted Moving Average. International Journal of Informatics and Information Systems, 6(2), 89–102.