

Implementasi Klasifikasi Teks Menggunakan Algoritma Naïve Bayes pada Sistem Pengarsipan Surat Masuk

Nur Afifah*, Ita Arfyanti², Yunita

Teknik Informatika, STMIK Widya Cipta Dharma, Samarinda, Indonesia

Email: ¹2143041@wicida.ac.id, ²*yunita@wicida.ac.id

Abstrak- Pengelolaan surat masuk merupakan bagian penting dalam administrasi perguruan tinggi karena berkaitan dengan penyimpanan, pengelompokan, dan pencarian kembali dokumen secara cepat dan akurat. Namun, proses pengarsipan surat masuk pada BAUK STMIK Widya Cipta Dharma masih dilakukan secara manual sehingga pengelompokan dokumen menjadi kurang efisien, membutuhkan waktu lebih lama dalam pencarian arsip, serta berpotensi menimbulkan ketidakkonsistenan dalam penentuan kategori surat. Penelitian ini bertujuan untuk mengimplementasikan algoritma Naïve Bayes pada sistem pengarsipan surat masuk berbasis web guna membantu proses klasifikasi surat secara otomatis. Sistem dikembangkan menggunakan Laravel dan Livewire sebagai aplikasi utama, serta Flask sebagai service klasifikasi. Data yang digunakan sebanyak 79 dokumen surat masuk yang terdiri atas empat kategori, yaitu permohonan, undangan, pemberitahuan, dan laporan. Tahap preprocessing meliputi case folding, cleaning text, tokenizing, stopword removal, dan stemming. Hasil penelitian menunjukkan bahwa sistem mampu melakukan klasifikasi surat masuk secara otomatis serta menampilkan detail proses klasifikasi melalui training report dan hasil klasifikasi. Berdasarkan pengujian terhadap 16 dokumen surat masuk, model memperoleh akurasi sebesar 75,00%, presisi rata-rata sebesar 63,89%, dan recall rata-rata sebesar 72,22%. Hasil tersebut menunjukkan bahwa algoritma Naïve Bayes cukup efektif dalam mendukung pengarsipan surat masuk secara lebih terstruktur dan efisien.

Kata Kunci: Naïve Bayes, pengarsipan surat masuk, klasifikasi dokumen, Laravel, Flask

Abstract- Incoming mail management is an important part of higher education administration because it is related to document storage, classification, and retrieval in a fast and accurate manner. However, the incoming mail archiving process at the General Administration and Finance Bureau (BAUK) of STMIK Widya Cipta Dharma is still carried out manually, making document grouping inefficient, slowing down archive retrieval, and potentially causing inconsistencies in determining mail categories. This study aims to implement the Naïve Bayes algorithm in a web-based incoming mail archiving system to support automatic mail classification. The system was developed using Laravel and Livewire as the main application, and Flask as the classification service. The dataset consisted of 79 incoming mail documents divided into four categories: requests, invitations, notifications, and reports. The preprocessing stage included case folding, text cleaning, tokenizing, stopword removal, and stemming. The results show that the system is able to automatically classify incoming mail and present detailed classification processes through training reports and classification results. Based on testing on 16 incoming mail documents, the model achieved an accuracy of 75.00%, an average precision of 63.89%, and an average recall of 72.22%. These results indicate that the Naïve Bayes algorithm is sufficiently effective in supporting a more structured and efficient incoming mail archiving process.

Keywords: Naïve Bayes, incoming mail archiving, document classification, Laravel, Flask

1. PENDAHULUAN

Perkembangan teknologi informasi telah membawa perubahan signifikan dalam berbagai bidang, termasuk dalam pengelolaan data dan dokumen pada lembaga pendidikan tinggi. Arsip merupakan sumber informasi yang memiliki fungsi penting dalam mendukung kegiatan administrasi, dokumentasi, serta pengambilan keputusan dalam suatu organisasi. Oleh karena itu, pengelolaan arsip dituntut untuk dilakukan secara tepat, cepat, dan akurat agar mampu menjamin kontinuitas kerja, kemudahan akses informasi, dan akuntabilitas administrasi [1],[2]. Pada era transformasi digital, sistem pengarsipan konvensional yang masih mengandalkan pencatatan manual tidak lagi memadai untuk memenuhi kebutuhan efisiensi dan kecepatan akses dokumen [3]. Pengarsipan elektronik menjadi salah satu solusi yang relevan karena mampu meningkatkan efektivitas pengelolaan dokumen sekaligus memperkuat transparansi dan akuntabilitas institusi [4],[5]. Salah satu unit yang memiliki kebutuhan tinggi terhadap pengelolaan arsip adalah Biro Administrasi Umum dan Keuangan (BAUK) STMIK Widya Cipta Dharma. Unit ini menangani berbagai dokumen administratif, seperti surat masuk, surat keluar, nota dinas, laporan, dan dokumen pendukung kegiatan operasional kampus. Namun, berdasarkan kondisi yang ada, proses pengarsipan surat masuk di BAUK masih dilakukan secara manual dan sebagian dokumen hanya disimpan dalam media digital sederhana, seperti layanan cloud storage. Walaupun penyimpanan berbasis cloud dapat membantu dalam penyimpanan dan berbagi dokumen, layanan tersebut belum memiliki kemampuan klasifikasi otomatis berdasarkan isi surat [6]. Akibatnya, proses pencarian, pengelompokan, dan penentuan jenis dokumen masih dilakukan secara manual, sehingga memerlukan waktu lebih lama dan berpotensi menimbulkan kesalahan pengarsipan. Kondisi ini menunjukkan bahwa sistem pengarsipan yang berjalan belum sepenuhnya mendukung efisiensi pengelolaan dokumen di lingkungan BAUK.

Untuk mengatasi permasalahan tersebut, diperlukan sistem pengarsipan cerdas yang mampu melakukan klasifikasi surat masuk secara otomatis berdasarkan isi dokumen. Salah satu metode yang dapat digunakan adalah algoritma Naïve Bayes. Algoritma ini merupakan metode klasifikasi probabilistik yang bekerja dengan menghitung peluang suatu dokumen termasuk ke dalam kategori tertentu berdasarkan distribusi kata yang muncul pada dokumen tersebut [7]. Naïve Bayes

dikenal memiliki struktur sederhana, mudah diimplementasikan, serta memiliki efisiensi komputasi yang baik pada proses klasifikasi data teks [8], [9]. Selain itu, algoritma ini efektif digunakan untuk klasifikasi dokumen karena mampu memanfaatkan pola kemunculan kata dalam menentukan kategori dokumen yang paling sesuai [10]. Karakteristik tersebut menjadikan Naïve Bayes relevan untuk diterapkan pada pengarsipan surat masuk yang berbasis atribut teks, seperti pengirim, perihal surat, dan isi surat [11].

Beberapa penelitian sebelumnya menunjukkan bahwa Naïve Bayes telah banyak diterapkan dalam berbagai kasus klasifikasi dokumen dan teks. Maharani, Ningrum, dan Utari menerapkan Naïve Bayes pada pengelolaan status retensi arsip dan memperoleh tingkat akurasi yang tinggi dalam menentukan status arsip [12]. Kasim, Yuyun, dan Sahibu mengembangkan sistem klasifikasi surat digital menggunakan Naïve Bayes dan TF-IDF dengan akurasi 86,24%, namun penelitian tersebut masih terbatas pada dua kategori surat [13]. Gerung, Maramis, dan Moningkey juga menerapkan Naïve Bayes pada sistem arsip surat berbasis web dan memperoleh akurasi 74,4%, tetapi fokus penelitian lebih menekankan pada pengembangan sistem daripada pendalaman karakteristik klasifikasi surat di institusi pendidikan [5]. Di lingkungan STMIK Widya Cipta Dharma sendiri, Yulindawati dkk. telah menunjukkan bahwa Naïve Bayes efektif digunakan untuk sistem rekomendasi pemilihan topik tugas akhir, yang menunjukkan potensi algoritma ini dalam menangani data kategorikal dan proses klasifikasi pada konteks akademik [14].

Meskipun demikian, penerapan Naïve Bayes dalam pengarsipan surat masuk di lingkungan perguruan tinggi, khususnya pada unit administrasi kampus, masih relatif terbatas. Sebagian besar penelitian terdahulu berfokus pada instansi pemerintahan, arsip retensi, atau sistem rekomendasi berbasis data akademik. Belum banyak penelitian yang secara khusus mengimplementasikan Naïve Bayes untuk mengklasifikasikan surat masuk berdasarkan isi teks pada konteks administrasi perguruan tinggi. Berdasarkan kondisi tersebut, penelitian ini dilakukan untuk mengimplementasikan algoritma Naïve Bayes dalam sistem pengarsipan dokumen surat masuk pada BAUK STMIK Widya Cipta Dharma.

Penelitian ini bertujuan untuk membangun sistem pengarsipan surat masuk berbasis web yang mampu melakukan klasifikasi dokumen secara otomatis ke dalam kategori tertentu berdasarkan isi surat. Selain menghasilkan sistem yang dapat digunakan dalam proses administrasi, penelitian ini juga diharapkan dapat meningkatkan efisiensi pengelolaan arsip, mempercepat pencarian dokumen, serta memberikan kontribusi akademik berupa penerapan algoritma Naïve Bayes pada pengarsipan surat masuk di lingkungan perguruan tinggi.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan model pengembangan Waterfall. Model ini dipilih karena memiliki tahapan yang sistematis dan terstruktur, sehingga sesuai untuk pengembangan sistem yang kebutuhan fungsionalnya telah dapat diidentifikasi sejak tahap awal [15]. Dalam penelitian ini, Waterfall digunakan untuk mengembangkan sistem pengarsipan surat masuk berbasis web yang dilengkapi dengan klasifikasi otomatis menggunakan algoritma Naïve Bayes. Tahapan penelitian meliputi analisis, desain, implementasi kode, dan pengujian sistem. Berikut merupakan alur penelitian yang dilakukan dengan menerapkan metode Waterfall.



Gambar 1. Alur Penelitian [14]

2.1 Analisis

Tahap analisis merupakan langkah awal dalam pengembangan sistem yang bertujuan untuk memahami kebutuhan pengguna, karakteristik data, serta algoritma klasifikasi yang akan diterapkan. Pada penelitian ini, tahap analisis dilakukan untuk mengidentifikasi permasalahan pada proses pengarsipan surat masuk di BAUK STMIK Widya Cipta Dharma, merumuskan kebutuhan sistem, menelaah karakteristik dataset surat masuk, serta menentukan alur penerapan algoritma Naïve Bayes dalam sistem. Tahap analisis terdiri atas empat bagian utama, yaitu analisis kebutuhan sistem, analisis data dan pengumpulan data, analisis algoritma Naïve Bayes, dan analisis text preprocessing.

2.1.1 Analisis Kebutuhan Sistem

Analisis kebutuhan dilakukan untuk mengidentifikasi permasalahan yang ada pada proses pengarsipan surat masuk di BAUK STMIK Widya Cipta Dharma. Berdasarkan hasil observasi dan wawancara dengan staf administrasi, diperoleh bahwa sistem pengarsipan yang berjalan masih bersifat manual dan sebagian dokumen hanya disimpan melalui Google Drive tanpa klasifikasi otomatis. Kebutuhan sistem terbagi menjadi kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional merupakan kebutuhan utama yang harus ada pada suatu sistem atau fungsi/fitur apa saja yang penting sehingga tidak dapat dipisahkan dari suatu sistem informasi, sedangkan kebutuhan non-fungsional merupakan kebutuhan pendukung dari suatu sistem yang dapat menunjang beroperasinya suatu sistem informasi [16]. Adapun kebutuhan fungsional sistem pada penelitian ini meliputi: Kebutuhan fungsional sistem yang dikembangkan meliputi kemampuan untuk menyimpan data surat masuk secara lengkap, yang mencakup nomor surat, tanggal surat, tanggal diterima, pengirim, perihal surat, isi surat, serta lampiran yang menyertainya. Sistem juga harus mampu menampilkan daftar surat masuk yang telah tersimpan sehingga memudahkan pengguna dalam melakukan pencarian dan pengelolaan data surat. Selain itu, sistem dapat melakukan klasifikasi otomatis terhadap surat masuk menggunakan algoritma Naïve Bayes dan menampilkan hasil klasifikasi dalam bentuk kategori surat yang sesuai.

Untuk mendukung transparansi proses klasifikasi, sistem menyediakan fitur yang dapat menampilkan detail proses klasifikasi, meliputi tahapan preprocessing teks, nilai probabilitas prior, kontribusi setiap token terhadap kategori yang dihasilkan, serta skor akhir pada masing-masing kelas. Sistem juga mampu menyajikan training report yang berfungsi membantu administrator dalam memahami proses pelatihan model klasifikasi. Selain itu, administrator diberikan hak untuk melakukan klasifikasi ulang maupun pelatihan ulang model apabila terdapat penambahan atau perubahan data latih sehingga performa model dapat terus ditingkatkan.

Sementara itu, kebutuhan nonfungsional sistem meliputi kemudahan penggunaan sehingga administrator dapat mengoperasikan sistem tanpa mengalami kesulitan yang berarti. Sistem juga mampu menyimpan data surat secara konsisten dan menjaga integritas data yang tersimpan. Dari aspek kinerja, sistem diharapkan memiliki waktu respons yang memadai ketika proses klasifikasi dilakukan sehingga pengguna dapat memperoleh hasil secara cepat. Dari aspek keamanan, sistem menerapkan pembatasan hak akses, di mana fitur klasifikasi dan pelatihan model hanya dapat dijalankan oleh administrator yang memiliki wewenang. Selain itu, sistem harus mampu menyajikan hasil klasifikasi secara akurat, jelas, dan mudah dipahami oleh pengguna sebagai dasar dalam pengelolaan surat masuk.

2.1.2 Analisis dan Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah dokumen surat masuk semester ganjil tahun akademik 2024/2025 yang diperoleh dari arsip BAUK STMIK Widya Cipta Dharma. Jumlah dataset yang digunakan sebanyak 79 dokumen surat masuk yang terdiri atas empat kategori, yaitu permohonan, undangan, pemberitahuan, dan laporan. Setiap dokumen direpresentasikan dengan tiga atribut utama, yaitu pengirim, perihal surat, dan isi surat, sedangkan label kategori digunakan sebagai target klasifikasi. Karena seluruh data yang tersedia dapat diolah secara penuh, penelitian ini tidak menggunakan teknik sampling, melainkan menerapkan total sampling atau sensus data. Pendekatan ini dipilih agar seluruh data yang tersedia dapat dimanfaatkan dalam proses pelatihan dan pengujian model, sehingga hasil klasifikasi lebih representatif terhadap kondisi riil surat masuk di BAUK STMIK Widya Cipta Dharma.

Pada tahap analisis data, dilakukan pemeriksaan awal terhadap struktur atribut dan konsistensi label kategori untuk memastikan kualitas data yang akan digunakan dalam proses pelatihan dan pengujian model klasifikasi. Kegiatan analisis ini diawali dengan penyesuaian nama atribut agar memiliki format yang seragam dan sesuai dengan kebutuhan sistem. Selanjutnya, dilakukan pemeriksaan terhadap kelengkapan data pada atribut penting, seperti pengirim, perihal surat, dan isi surat, guna memastikan tidak terdapat data yang kosong atau tidak valid.

Tahap berikutnya adalah penyeragaman penulisan kategori surat untuk menghindari perbedaan penamaan yang dapat memengaruhi hasil klasifikasi. Selain itu, dilakukan perbaikan terhadap kesalahan penulisan (typo) pada label kategori sehingga setiap data memiliki label yang konsisten dan akurat. Setelah proses pembersihan dan validasi data selesai, data kemudian dipisahkan berdasarkan kategori yang telah ditentukan. Pemisahan ini bertujuan untuk mempersiapkan data yang akan digunakan pada proses pelatihan (training) dan evaluasi (testing) model klasifikasi, sehingga kinerja model dapat diukur secara objektif dan menghasilkan tingkat akurasi yang lebih baik.

Dengan demikian, hasil tahap ini adalah dataset surat masuk yang telah bersih, konsisten, dan siap diproses pada tahap text preprocessing serta klasifikasi.

2.1.3 Analisis Algoritma Naïve Bayes

Penelitian ini menggunakan algoritma Naïve Bayes untuk melakukan klasifikasi otomatis terhadap dokumen surat masuk. Naïve Bayes merupakan algoritma klasifikasi probabilistik yang menghitung peluang suatu dokumen termasuk ke dalam kelas tertentu berdasarkan distribusi kata-kata yang muncul pada dokumen tersebut [17], [18]. Metode ini dipilih karena memiliki struktur sederhana, mudah diimplementasikan, dan efisien secara komputasi untuk menangani data berbasis teks [8], [9].

Pada penelitian ini, setiap dokumen surat masuk dianalisis berdasarkan atribut pengirim, perihal surat, dan isi surat. Naïve Bayes menghitung probabilitas masing-masing kelas dengan menggabungkan *prior* dan *likelihood token* yang muncul pada dokumen. Untuk menghindari probabilitas nol ketika terdapat *token* yang belum pernah muncul pada data *training*, penelitian ini menggunakan *Laplace smoothing*. Probabilitas awal tiap kelas dihitung menggunakan rumus:

$$P(C_i) = \frac{N_i}{N} \quad (1)$$

Keterangan:

$P(C_i)$ = probabilitas awal kelas ke- i

N_i = jumlah dokumen pada kelas ke- i

N = jumlah seluruh dokumen *training*

Probabilitas token pada suatu kelas dihitung dengan rumus:

$$P(w_k | C_i) = \frac{n_{ik} + 1}{\sum n_i + |V|} \quad (2)$$

Keterangan:

$P(w_k | C_i)$ = probabilitas token w_k pada kelas C_i

n_{ik} = Jumlah kemunculan token w_k pada kelas C_i

$\sum n_{ik}$ = total token pada kelas C_i

$|V|$ = jumlah *vocabulary*

Untuk menentukan hasil klasifikasi, nilai log prior dan log likelihood dari seluruh token dijumlahkan sehingga diperoleh skor total tiap kelas:

$$\log P(C_i | d) = \log P(C_i) + \sum_{k=1}^n \log P(w_k | C_i) \quad (3)$$

Dokumen kemudian diklasifikasikan ke dalam kelas dengan nilai skor total terbesar. Penggunaan bentuk logaritmik dilakukan untuk menjaga stabilitas perhitungan dan menghindari underflow akibat perkalian banyak probabilitas kecil.

2.1.4 Analisis Text Pre-Processing

Dokumen kemudian diklasifikasikan ke dalam kelas dengan nilai skor total terbesar. Penggunaan bentuk logaritmik dilakukan untuk menjaga stabilitas perhitungan dan menghindari *underflow* akibat perkalian banyak probabilitas kecil. Pada penelitian ini, *text preprocessing* yang diterapkan terdiri atas beberapa tahap berikut:

a. *Case folding*

Case folding merupakan proses mengubah seluruh huruf pada teks menjadi huruf kecil. Tahap ini dilakukan agar kata yang sama tidak dianggap berbeda hanya karena perbedaan huruf kapital.

b. *Cleaning Text*

Cleaning text dilakukan dengan menghapus tanda baca, karakter khusus, dan spasi berlebih yang tidak memberikan makna penting dalam proses klasifikasi. Tahap ini menghasilkan teks yang lebih bersih dan lebih seragam untuk diproses lebih lanjut.

c. *Tokenizing*

Tokenizing merupakan proses memecah teks menjadi unit-unit kata atau token [18]. Pada penelitian ini, tokenisasi diterapkan pada atribut pengirim, perihal surat, dan isi surat. Token hasil pemecahan kata inilah yang menjadi dasar dalam perhitungan probabilitas Naïve Bayes.

d. *Filtering*

Filtering digunakan untuk menyaring kata-kata yang bersifat umum dan kurang memiliki kontribusi terhadap pembeda kategori, seperti kata sambung atau kata umum lainnya [18]. Pada implementasi sistem, *filtering* diterapkan

secara terbatas untuk menjaga agar token yang diproses lebih relevan terhadap isi surat, tanpa menghilangkan kata-kata penting yang justru dapat menjadi penanda kategori.

e. *Stemming*

Stemming merupakan proses mengubah kata berimbuhan menjadi bentuk dasar [19]. Berdasarkan penelitian ini, *stemming* dipertimbangkan sebagai langkah untuk menyeragamkan variasi kata yang memiliki akar makna sama. Namun, penerapan *stemming* dilakukan secara selektif agar tidak mengubah makna administratif yang spesifik pada surat. Hasil akhir dari tahap *preprocessing* adalah kumpulan token yang telah dibersihkan dan distandardisasi. Token-token tersebut kemudian digunakan sebagai dasar dalam menghitung *prior*, *likelihood*, dan skor klasifikasi pada algoritma Naïve Bayes [20].

2.2 Design

Tahap desain bertujuan untuk menerjemahkan hasil analisis kebutuhan ke dalam bentuk rancangan sistem yang lebih terstruktur. Pada tahap ini dilakukan perancangan arsitektur perangkat lunak, basis data, dan antarmuka pengguna.

2.2.1 Perancangan Arsitektur Sistem

Sistem dirancang menggunakan arsitektur berbasis web dengan pendekatan pemisahan modul. Aplikasi utama dibangun menggunakan Laravel dan Livewire untuk pengelolaan data surat masuk dan antarmuka pengguna, sedangkan proses klasifikasi dibangun sebagai *service* terpisah menggunakan Flask. Komunikasi antara aplikasi utama dan *service* klasifikasi dilakukan melalui API berbasis HTTP. Arsitektur ini dipilih agar proses pengelolaan data surat dan proses komputasi klasifikasi dapat dipisahkan secara modular. Dengan demikian, pengembangan logika klasifikasi dapat dilakukan tanpa mengganggu modul utama pengarsipan.

2.2.2 Perancangan Basis Data

Basis data dirancang untuk menyimpan data surat beserta hasil klasifikasinya. Tabel utama menyimpan informasi nomor surat, tanggal surat, tanggal terima, pengirim, perihal surat, isi surat, lampiran, status klasifikasi, kategori hasil klasifikasi, skor terbaik, dan hasil klasifikasi dalam format JSON. Struktur ini dirancang agar sistem tidak hanya menyimpan hasil akhir klasifikasi, tetapi juga mampu menampilkan kembali detail proses klasifikasi untuk keperluan audit dan penjabaran.

2.2.3 Perancangan Antarmuka

Antarmuka sistem dirancang untuk mendukung dua kebutuhan utama. Pertama, halaman pengelolaan surat masuk yang digunakan untuk menyimpan dan menampilkan data surat. Kedua, halaman admin klasifikasi yang digunakan untuk melakukan *training model*, menjalankan klasifikasi, melihat hasil klasifikasi, serta menampilkan detail tahapan perhitungan dan *training report*.

2.3 Kode

Tahap implementasi kode merupakan proses menerjemahkan rancangan sistem ke dalam bentuk program yang dapat dijalankan. Aplikasi utama dikembangkan menggunakan Laravel sebagai *framework backend* dan Livewire untuk membangun antarmuka dinamis pada sisi aplikasi web. Basis data yang digunakan adalah MySQL, sedangkan service klasifikasi dibangun menggunakan Flask dengan bahasa pemrograman Python.

Pada implementasi ini, Laravel berfungsi untuk menangani proses CRUD data surat masuk, autentikasi, penyimpanan lampiran, dan tampilan antarmuka. Sementara itu, Flask berfungsi untuk:

- Melatih model Naïve Bayes.
- Menerima data surat baru untuk diklasifikasikan.
- Menghitung hasil klasifikasi.
- Mengembalikan hasil klasifikasi beserta penjabaran proses perhitungan.

Integrasi antara Laravel dan Flask dilakukan melalui *request API*. Ketika admin menjalankan proses klasifikasi, data surat dikirim dari aplikasi Laravel ke Flask, kemudian Flask mengembalikan hasil berupa kategori prediksi, skor terbaik, ranking kelas, *preprocessing*, dan penjelasan proses klasifikasi. Hasil tersebut selanjutnya disimpan ke basis data dan ditampilkan kembali pada halaman admin.

2.4 Testing

Tahap pengujian merupakan bagian akhir dari proses pengembangan sistem yang bertujuan untuk memastikan bahwa algoritma Naïve Bayes yang diterapkan mampu melakukan klasifikasi dokumen surat masuk dengan hasil yang akurat. Pengujian akurasi sistem dilakukan dengan menggunakan Confusion Matrix untuk mengukur tingkat ketepatan hasil klasifikasi algoritma Naïve Bayes terhadap data surat masuk. Data uji yang digunakan sebanyak 81 dokumen surat, masing-masing telah diberi label kategori aktual berdasarkan isi surat oleh peneliti. Sistem kemudian mengklasifikasikan setiap dokumen, dan hasil klasifikasi dibandingkan dengan label sebenarnya untuk menghasilkan nilai True Positive (TP),

False Positive (FP), False Negative (FN), dan True Negative (TN). Pengukuran utama kinerja klasifikasi adalah Akurasi (Accuracy, AC), yang didefinisikan sebagai rasio antara jumlah kasus yang berhasil diklasifikasikan dengan benar (correctly classified instances) dalam test set dengan jumlah total kasus dalam test set. Selain Akurasi, metrik Presisi (Precision, P) dan Recall (R) digunakan untuk mengukur ketepatan (exactness) dan kelengkapan (completeness) klasifikasi, Metrik tersebut dapat dihitung menggunakan rumus-rumus sebagai berikut:

$$AC = \frac{TP + T}{TP + TN + FN + FP} \quad (5)$$

$$R = \frac{TP}{FN + TP} \quad (6)$$

$$P = \frac{TP}{FP + TP} \quad (7)$$

Keterangan:

TP: Jumlah prediksi yang tepat bahwa kelas bersifat True Positif

TN: Jumlah prediksi yang tepat bahwa kelas bersifat True Negative

FP: Jumlah prediksi yang salah bahwa kelas bersifat False Positive

FN: Jumlah prediksi yang salah bahwa kelas bersifat False Negative

Selain pengujian model, dilakukan pula pengujian fungsional sistem untuk memastikan bahwa fitur-fitur utama berjalan dengan baik, meliputi penyimpanan surat masuk, klasifikasi surat, klasifikasi ulang, penampilan hasil klasifikasi, serta penampilan training report.

3. HASIL DAN PEMBAHASAN

3.1 Rancangan Pengembangan Sistem

Hasil penelitian ini berupa sistem pengarsipan dokumen surat masuk berbasis web yang dilengkapi dengan fitur klasifikasi otomatis menggunakan algoritma Naïve Bayes. Sistem ini dikembangkan untuk membantu BAUK STMIK Widya Cipta Dharma dalam mengelola surat masuk secara lebih terstruktur, cepat, dan efisien. Pengembangan sistem dilakukan berdasarkan kebutuhan yang telah dianalisis sebelumnya, yaitu kebutuhan untuk menyimpan data surat, mengelompokkan surat secara otomatis berdasarkan isi dokumen, serta menampilkan penjabaran hasil klasifikasi agar proses pengambilan keputusan lebih transparan.

Secara umum, sistem dibangun dengan arsitektur modular yang memisahkan aplikasi utama dengan modul klasifikasi. Aplikasi utama dikembangkan menggunakan framework Laravel dan Livewire untuk menangani proses pengelolaan data surat masuk, autentikasi pengguna, dan tampilan antarmuka. Sementara itu, modul klasifikasi dikembangkan menggunakan Flask yang berfungsi sebagai layanan terpisah untuk melakukan proses training model dan klasifikasi surat. Komunikasi antara Laravel dan Flask dilakukan melalui Application Programming Interface (API), sehingga data surat dapat dikirim dari sistem utama ke service klasifikasi dan hasilnya dapat dikembalikan ke aplikasi utama.

Rancangan sistem ini dipilih karena memberikan beberapa keuntungan. Pertama, pemisahan modul membuat pengembangan aplikasi utama dan model klasifikasi menjadi lebih terstruktur. Kedua, perawatan sistem menjadi lebih mudah karena perubahan pada logika klasifikasi tidak secara langsung memengaruhi modul pengelolaan surat. Ketiga, pendekatan ini memungkinkan sistem untuk menampilkan hasil klasifikasi secara rinci, termasuk tahapan preprocessing, ringkasan training model, token dominan, dan skor klasifikasi setiap kelas.

Secara fungsional, sistem memiliki dua bagian utama. Bagian pertama adalah modul pengelolaan surat masuk, yang digunakan untuk menyimpan, menampilkan, dan mengelola data surat. Bagian kedua adalah modul admin klasifikasi, yang digunakan untuk melatih model, menjalankan klasifikasi, melihat detail hasil klasifikasi, serta menampilkan training report. Dengan rancangan ini, sistem tidak hanya berperan sebagai media penyimpanan dokumen, tetapi juga sebagai alat bantu pengambilan keputusan dalam pengarsipan surat masuk.

3.2 Flowchart dan UML Sistem

3.2.1 Flowcart Sistem

Pengarsipan dan klasifikasi surat masuk. Proses dimulai ketika admin melakukan login ke dalam sistem, kemudian menginput data surat masuk berupa nomor surat, tanggal surat, tanggal terima, pengirim, perihal surat, isi surat, dan lampiran. Setelah data berhasil disimpan ke dalam basis data, admin dapat memilih untuk menjalankan proses klasifikasi terhadap surat yang diinginkan.

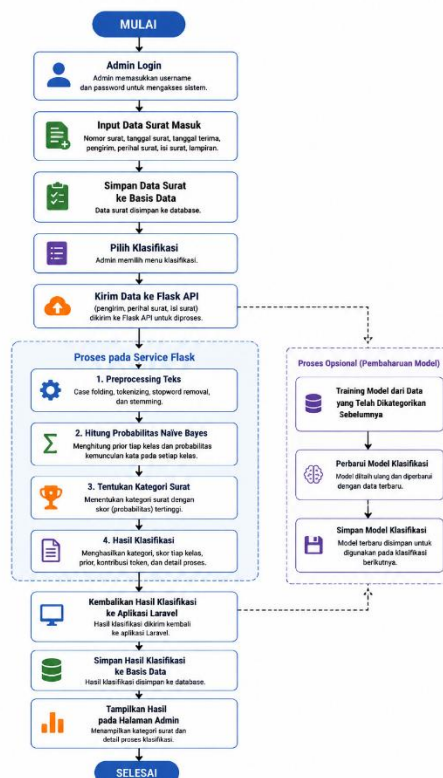
Pada tahap klasifikasi, sistem mengambil atribut teks surat berupa pengirim, perihal surat, dan isi surat, kemudian mengirimkannya ke service Flask melalui API. Service Flask menerima data tersebut dan memulai tahap preprocessing teks yang meliputi case folding, cleaning text, tokenizing, stopword removal, dan stemming. Setelah preprocessing

selesai, sistem menghitung probabilitas setiap kelas menggunakan algoritma Naïve Bayes, kemudian menentukan kategori surat dengan skor tertinggi sebagai hasil klasifikasi.

Hasil klasifikasi yang diperoleh dari service Flask dikembalikan ke aplikasi Laravel untuk disimpan ke basis data. Selanjutnya, hasil klasifikasi ditampilkan pada halaman admin dalam bentuk kategori hasil, skor terbaik, serta detail penjabaran proses klasifikasi. Selain proses klasifikasi, flowchart juga menunjukkan adanya proses opsional berupa pelatihan ulang model menggunakan data surat yang telah memiliki label kategori. Proses ini dilakukan untuk memperbarui model agar sesuai dengan perkembangan data yang tersedia.

Berdasarkan alur tersebut, *flowchart* sistem menunjukkan bahwa proses pengarsipan dan klasifikasi berjalan secara terintegrasi, mulai dari input data hingga penampilan hasil pada antarmuka admin. Gambar 2 menunjukkan flowchart sistem pengarsipan dan klasifikasi surat masuk.

Flowchart Sistem Pengelolaan dan Klasifikasi Surat Masuk



Gambar 2. Flowchart Sistem

3.2.2 UML Sistem

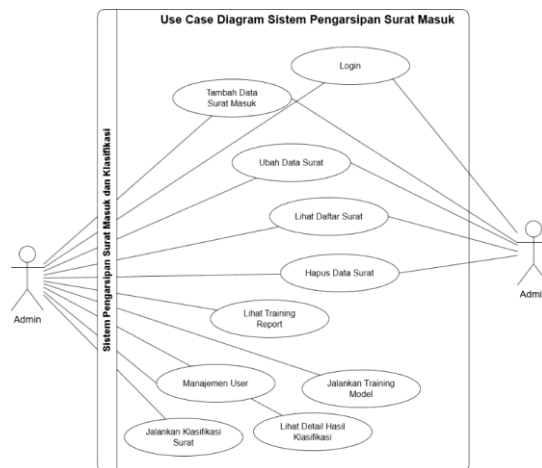
Untuk menggambarkan struktur dan interaksi sistem, penelitian ini menggunakan UML berupa *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*.

a. Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan hubungan antara aktor dengan fungsi-fungsi utama sistem. Pada penelitian ini, aktor utama adalah admin, karena seluruh fitur pengelolaan surat dan klasifikasi hanya dapat diakses oleh pengguna yang memiliki hak akses administratif. Admin berinteraksi langsung dengan sistem untuk mengelola data surat masuk, menjalankan proses klasifikasi, melakukan training model, dan melihat hasil analisis model.

Berdasarkan Use Case Diagram, admin dapat melakukan beberapa aktivitas utama, yaitu login, menambah data surat masuk, melihat daftar surat, mengubah data surat, menghapus data surat, menjalankan klasifikasi surat, melakukan klasifikasi ulang, menjalankan training model, melihat training report, serta melihat detail hasil klasifikasi. Diagram ini menunjukkan bahwa sistem tidak hanya berperan sebagai aplikasi arsip, tetapi juga menyediakan fasilitas analisis yang mendukung proses klasifikasi secara transparan.

Adanya fitur detail hasil klasifikasi dan training report menjadi pembeda penting dari sistem yang dibangun, karena admin dapat melihat bukan hanya hasil akhir, tetapi juga proses dan data pendukung yang digunakan model. Dengan demikian, Use Case Diagram memperlihatkan bahwa seluruh fungsi sistem dirancang untuk mendukung pengelolaan arsip yang lebih terstruktur, terintegrasi, dan informatif. Gambar 3 menunjukkan Use Case Diagram sistem pengarsipan surat masuk.



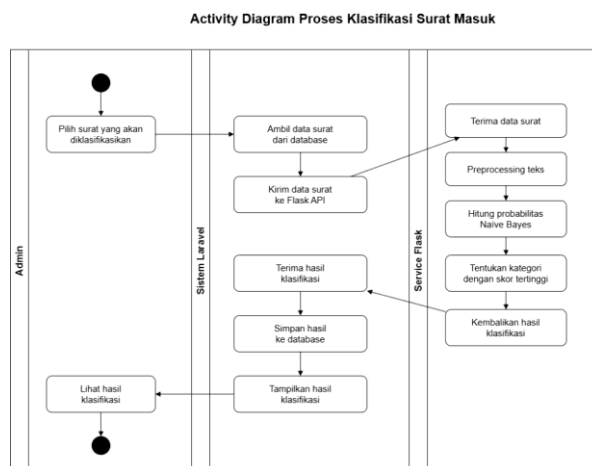
Gambar 3. Use Case Diagram

Use Case Diagram pada Gambar 3 menggambarkan hubungan antara admin dengan seluruh fungsi utama yang tersedia pada sistem pengarsipan surat masuk. Dalam diagram tersebut, admin berperan sebagai aktor utama yang memiliki hak akses penuh terhadap fitur-fitur sistem. Aktivitas yang dapat dilakukan admin meliputi login, pengelolaan data surat masuk, menjalankan proses klasifikasi, melakukan klasifikasi ulang, menjalankan training model, melihat training report, serta melihat detail hasil klasifikasi. Diagram ini menunjukkan bahwa sistem yang dibangun tidak hanya berfungsi sebagai media penyimpanan arsip surat, tetapi juga mendukung proses analisis dan klasifikasi dokumen secara otomatis. Integrasi fitur klasifikasi dan training model pada satu sistem memberikan kemudahan bagi admin dalam melakukan pengelolaan arsip secara lebih terstruktur dan efisien. Selain itu, adanya fitur detail hasil klasifikasi dan training report memberikan transparansi terhadap proses yang dilakukan oleh algoritma Naïve Bayes sehingga hasil klasifikasi dapat dipahami dengan lebih jelas oleh pengguna.

b. Activity Diagram

Activity Diagram digunakan untuk menjelaskan alur aktivitas pada salah satu proses utama dalam sistem, yaitu proses klasifikasi surat masuk. Proses dimulai ketika admin memilih surat yang akan diklasifikasikan. Sistem kemudian mengambil data surat dari basis data dan menyiapkan atribut yang diperlukan, yaitu pengirim, perihal surat, dan isi surat. Selanjutnya, sistem Laravel mengirimkan data tersebut ke service Flask. Setelah menerima data surat, service Flask melakukan preprocessing teks dan menghitung probabilitas kelas menggunakan algoritma Naïve Bayes. Berdasarkan hasil perhitungan, Flask menentukan kategori dengan skor tertinggi sebagai hasil akhir klasifikasi. Hasil ini kemudian dikembalikan ke Laravel untuk disimpan ke basis data.

Setelah hasil klasifikasi tersimpan, sistem menampilkan kategori dan detail analisis pada halaman admin. Activity Diagram ini menunjukkan bahwa proses klasifikasi berlangsung melalui interaksi antara pengguna, aplikasi utama, dan service klasifikasi. Pemisahan aktivitas tersebut memperlihatkan bahwa sistem dirancang secara modular, di mana aplikasi utama bertugas menangani pengelolaan data dan tampilan, sementara service Flask menangani logika klasifikasi. Gambar 4 menunjukkan Activity Diagram proses klasifikasi surat masuk.



Gambar 4. Activity Diagram

Activity Diagram pada Gambar 4 menggambarkan alur aktivitas sistem pada proses klasifikasi surat masuk. Proses dimulai ketika admin memilih surat yang akan diklasifikasikan. Sistem kemudian mengambil data surat dari basis data, khususnya atribut pengirim, perihal surat, dan isi surat, untuk dikirimkan ke service Flask melalui API. Setelah data diterima, service Flask melakukan tahap preprocessing yang meliputi case folding, cleaning text, tokenizing, stopword removal, dan stemming. Selanjutnya, sistem menghitung probabilitas setiap kategori menggunakan algoritma Naïve Bayes untuk menentukan kategori surat dengan skor tertinggi. Hasil klasifikasi kemudian dikembalikan ke aplikasi Laravel untuk disimpan ke dalam basis data dan ditampilkan pada halaman admin. Diagram ini menunjukkan bahwa proses klasifikasi berjalan secara terstruktur melalui interaksi antara pengguna, aplikasi utama, dan service klasifikasi. Pemisahan proses antara Laravel dan Flask juga memperlihatkan bahwa sistem dirancang secara modular sehingga memudahkan pengembangan dan pemeliharaan sistem di masa mendatang.

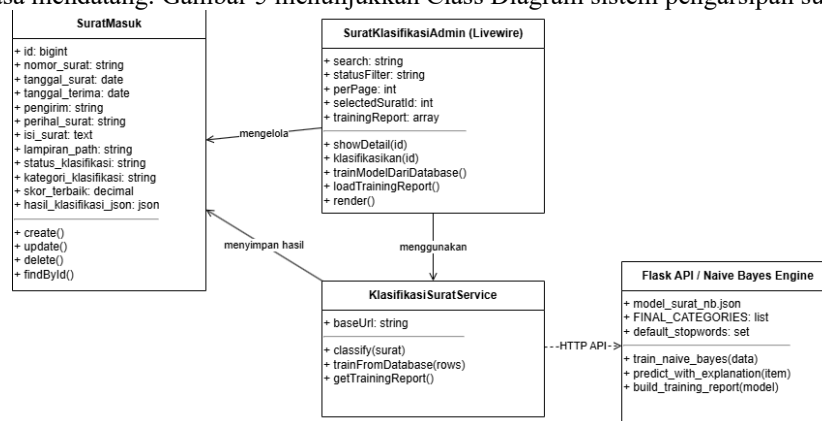
c. *Class Diagram*

Class Diagram digunakan untuk menggambarkan struktur kelas utama yang membentuk sistem pengarsipan surat masuk. Diagram ini menunjukkan entitas data utama, komponen antarmuka, *service* klasifikasi, serta hubungan antar komponen tersebut.

Kelas utama dalam sistem adalah SuratMasuk, yang merepresentasikan data surat masuk dan hasil klasifikasinya. Kelas ini memiliki atribut seperti nomor surat, tanggal surat, tanggal terima, pengirim, perihal surat, isi surat, lampiran, status klasifikasi, kategori hasil klasifikasi, skor terbaik, dan hasil klasifikasi dalam format JSON. Kelas ini menjadi pusat penyimpanan informasi surat yang digunakan oleh sistem.

Selain itu, terdapat komponen SuratKlasifikasiAdmin yang dibangun menggunakan Livewire untuk menangani interaksi pengguna pada halaman admin klasifikasi. Komponen ini berperan dalam menampilkan daftar surat, menampilkan detail hasil klasifikasi, menjalankan klasifikasi ulang, dan memuat training report. Komponen tersebut menggunakan KlasifikasiSuratService sebagai penghubung antara aplikasi Laravel dengan *service* Flask. Sementara itu, di sisi Flask terdapat modul yang menjalankan proses training model dan klasifikasi menggunakan algoritma Naïve Bayes.

Class Diagram ini menunjukkan bahwa sistem dibangun dengan pemisahan tanggung jawab yang jelas antar komponen. Struktur tersebut mendukung pengembangan aplikasi yang lebih terorganisasi, mudah dirawat, dan memungkinkan perluasan fitur di masa mendatang. Gambar 5 menunjukkan Class Diagram sistem pengarsipan surat masuk.



Gambar 5. *Class diagram*

Class Diagram pada Gambar 5 digunakan untuk menggambarkan struktur kelas utama yang membentuk sistem pengarsipan surat masuk. Diagram ini memperlihatkan hubungan antar kelas, atribut yang digunakan, serta peran masing-masing komponen dalam sistem. Kelas utama pada sistem adalah SuratMasuk yang berfungsi untuk menyimpan data surat beserta hasil klasifikasinya. Kelas ini memiliki atribut seperti nomor surat, tanggal surat, tanggal terima, pengirim, perihal surat, isi surat, lampiran, status klasifikasi, kategori hasil klasifikasi, skor terbaik, dan hasil klasifikasi dalam format JSON. Selain itu, terdapat komponen SuratKlasifikasiAdmin yang berfungsi menangani interaksi pengguna pada halaman admin, seperti menjalankan klasifikasi, menampilkan hasil klasifikasi, dan melihat training report. Diagram ini juga menunjukkan adanya KlasifikasiSuratService sebagai penghubung antara aplikasi Laravel dengan service Flask. Struktur tersebut memperlihatkan bahwa sistem dibangun dengan pemisahan tanggung jawab yang jelas sehingga setiap komponen memiliki fungsi yang spesifik dan saling mendukung dalam proses pengarsipan serta klasifikasi surat masuk.

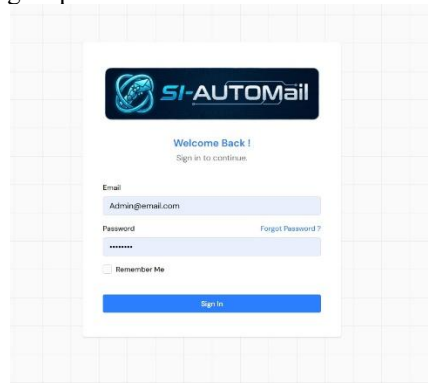
3.3 Implementasi Aplikasi

3.3.1 Halaman Login

Implementasi aplikasi diawali dengan penyediaan halaman login sebagai pintu masuk pengguna ke dalam sistem. Halaman ini digunakan untuk membatasi akses ke fitur-fitur utama, terutama fitur pengelolaan surat dan klasifikasi yang

hanya dapat digunakan oleh admin. Melalui halaman login, pengguna diminta memasukkan email dan password agar dapat masuk ke sistem.

Keberadaan halaman login penting untuk menjaga keamanan data arsip dan memastikan bahwa proses pengelolaan surat dilakukan oleh pihak yang berwenang. Dari sisi implementasi, halaman ini juga menjadi bagian awal dari alur sistem, karena seluruh proses pengelolaan surat dan klasifikasi baru dapat dilakukan setelah autentikasi berhasil. Gambar 6 menunjukkan halaman login sistem pengarsipan surat masuk.



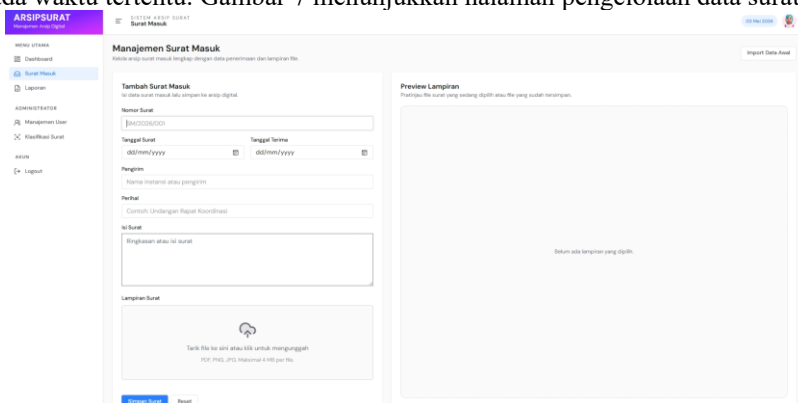
Gambar 6. Halaman Login

Halaman login pada Gambar 6 merupakan tampilan awal yang digunakan pengguna untuk masuk ke dalam sistem. Halaman ini berfungsi sebagai mekanisme autentikasi agar hanya pengguna yang memiliki hak akses yang dapat menggunakan fitur-fitur utama pada aplikasi. Melalui halaman login, admin diminta memasukkan email dan password sebelum dapat mengakses sistem pengarsipan surat masuk. Implementasi halaman login menjadi bagian penting dalam menjaga keamanan data arsip serta membatasi akses terhadap fitur klasifikasi dan training model yang hanya diperuntukkan bagi admin.

3.3.2 Halaman Pengelolaan Surat Masuk

Setelah berhasil login, admin dapat mengakses halaman pengelolaan surat masuk. Halaman ini berfungsi untuk menambah, menampilkan, mengubah, dan menghapus data surat masuk yang diterima oleh BAUK STMIK Widya Cipta Dharma. Data yang dikelola pada halaman ini meliputi nomor surat, tanggal surat, tanggal terima, pengirim, perihal surat, isi surat, dan lampiran dokumen.

Halaman ini merupakan modul dasar dari aplikasi karena seluruh data yang akan diproses oleh algoritma Naïve Bayes berasal dari data yang disimpan di sini. Dengan adanya modul pengelolaan surat masuk, admin dapat mendokumentasikan surat secara lebih terstruktur dan konsisten. Selain itu, modul ini juga mempermudah proses pencarian kembali arsip apabila dibutuhkan pada waktu tertentu. Gambar 7 menunjukkan halaman pengelolaan data surat masuk.

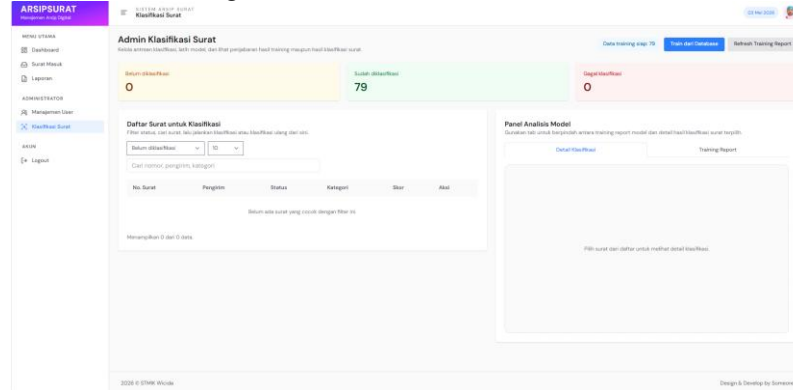


Gambar 7. Halaman Pengelolaan Surat Masuk

Gambar 6 menunjukkan halaman pengelolaan surat masuk yang digunakan admin untuk mengelola data surat secara langsung. Pada halaman ini tersedia fitur untuk menambah, menampilkan, mengubah, dan menghapus data surat masuk. Data yang dikelola meliputi nomor surat, tanggal surat, tanggal terima, pengirim, perihal surat, isi surat, dan lampiran dokumen. Halaman ini menjadi bagian penting dalam sistem karena seluruh data surat yang digunakan pada proses klasifikasi berasal dari modul ini. Dengan adanya pengelolaan data yang terstruktur, proses penyimpanan dan pencarian kembali arsip surat menjadi lebih mudah dan efisien dibandingkan pengarsipan manual.

3.3.3 Halaman Admin Klasifikasi Surat

Halaman admin klasifikasi merupakan inti dari aplikasi yang dibangun. Pada halaman ini, admin dapat melihat jumlah surat berdasarkan status klasifikasinya, seperti belum diklasifikasi, sudah diklasifikasi, dan gagal diklasifikasi. Selain itu, halaman ini juga menampilkan daftar surat yang dapat dipilih untuk diproses, serta menyediakan tombol untuk menjalankan klasifikasi dan melakukan training model.



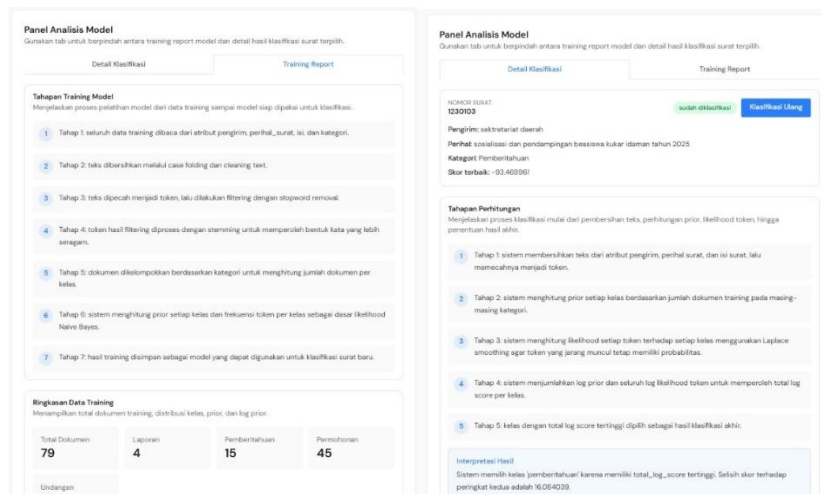
Gambar 8. Halaman Klasifikasi Surat

Implementasi halaman ini menunjukkan integrasi langsung antara aplikasi pengarsipan dengan modul klasifikasi. Seluruh proses dilakukan melalui satu antarmuka, sehingga admin tidak perlu berpindah ke sistem lain untuk menjalankan klasifikasi. Fitur ini membuat proses pengarsipan dan analisis surat menjadi lebih praktis dan terpusat. Gambar 8 menunjukkan halaman admin klasifikasi surat masuk.

3.3.4 Tab Training Report dan Hasil Klasifikasi Surat

Salah satu fitur penting yang diimplementasikan adalah training report. Tab ini digunakan untuk menampilkan hasil pelatihan model Naïve Bayes dalam bentuk yang mudah dipahami oleh admin. Informasi yang disajikan meliputi tahapan training model, jumlah data training, distribusi kelas, nilai prior, log prior, ukuran vocabulary, dan token dominan pada masing-masing kategori. *Training report* memiliki fungsi penting karena memberikan gambaran mengenai kondisi model yang digunakan. Dengan melihat distribusi kelas dan token dominan, admin dapat memahami apakah data training yang tersedia sudah cukup representatif atau masih perlu diperbaiki. Fitur ini juga memperkuat transparansi sistem, karena model yang digunakan tidak lagi bersifat tertutup bagi pengguna.

Selain *training report*, sistem juga menyediakan tab detail hasil klasifikasi yang menampilkan penjabaran lebih rinci terhadap satu surat yang dipilih. Pada tab ini, sistem menampilkan kategori hasil klasifikasi, skor terbaik, tahapan perhitungan, preprocessing teks, skor tiap kelas, dan ranking akhir seluruh kategori. Implementasi fitur ini memberikan keunggulan penting, karena pengguna dapat memahami alasan di balik hasil klasifikasi yang diberikan sistem. Dengan kata lain, sistem tidak hanya menampilkan label akhir, tetapi juga menunjukkan proses statistik dan tekstual yang mendasari keputusan tersebut. Hal ini membuat sistem lebih transparan dan lebih mudah dipertanggungjawabkan dalam konteks administrasi. Gambar 8 menunjukkan tampilan detail hasil klasifikasi surat masuk dan training report model Naïve Bayes.



Gambar 9. Tab Training Report dan Hasil Klasifikasi Surat

Gambar 9 menunjukkan tampilan training report dan detail hasil klasifikasi surat. Fitur training report digunakan untuk menampilkan informasi mengenai proses pelatihan model Naïve Bayes, seperti jumlah data training, distribusi kelas, nilai prior, log prior, ukuran vocabulary, serta token dominan pada masing-masing kategori. Sementara itu, tab detail hasil klasifikasi digunakan untuk menampilkan penjabaran proses klasifikasi terhadap surat yang dipilih. Informasi yang ditampilkan meliputi hasil preprocessing, skor tiap kelas, ranking kategori, dan hasil akhir klasifikasi. Implementasi fitur ini menjadi salah satu keunggulan sistem karena hasil klasifikasi tidak hanya ditampilkan dalam bentuk kategori akhir, tetapi juga disertai penjelasan proses perhitungan yang dilakukan oleh model. Dengan adanya transparansi tersebut, admin dapat memahami alasan sistem dalam menentukan kategori suatu surat sehingga hasil klasifikasi menjadi lebih mudah dipahami dan dipertanggungjawabkan.

3.4 Implementasi Algoritma Naïve Bayes

Implementasi algoritma Naïve Bayes dilakukan pada service Flask yang terpisah dari aplikasi utama. Service ini bertanggung jawab untuk memproses data training, melakukan preprocessing teks, menghitung probabilitas kelas, serta mengembalikan hasil klasifikasi ke aplikasi Laravel. Pada penelitian ini, model digunakan untuk mengklasifikasikan surat masuk ke dalam empat kategori, yaitu permohonan, undangan, pemberitahuan, dan laporan.

3.4.1 Tahap *Preprocessing*

Sebelum dilakukan klasifikasi, data teks surat terlebih dahulu diproses melalui tahap preprocessing. Pada penelitian ini, preprocessing mencakup beberapa langkah, yaitu case folding, cleaning text, tokenizing, stopword removal, dan stemming. Case folding dilakukan dengan mengubah seluruh huruf menjadi huruf kecil agar tidak terjadi perbedaan token hanya karena kapitalisasi. Cleaning text dilakukan untuk membersihkan tanda baca, karakter khusus, dan spasi berlebih. Setelah itu, teks dipecah menjadi token-token kata melalui proses tokenizing.

Tahap berikutnya adalah stopword removal, yaitu penghapusan kata-kata umum yang memiliki kontribusi rendah dalam membedakan kategori surat. Setelah filtering selesai, token yang tersisa diproses menggunakan stemming untuk mengubah kata berimbuhan menjadi bentuk dasarnya. Dengan preprocessing ini, token yang digunakan dalam proses klasifikasi menjadi lebih bersih, seragam, dan representatif terhadap isi dokumen.

Penerapan preprocessing dalam sistem tidak hanya digunakan untuk proses komputasi, tetapi juga ditampilkan pada halaman admin dalam bentuk penjabaran bertahap. Hal ini membuat admin dapat melihat perubahan teks dari bentuk asli hingga menjadi token final yang dipakai model.

3.4.2 Proses *Training Model*

Training model dilakukan dengan memanfaatkan data surat masuk yang telah memiliki label kategori. Pada tahap ini, sistem membaca data training dari basis data, kemudian menerapkan preprocessing pada atribut pengirim, perihal surat, dan isi surat. Setelah token diperoleh, sistem menghitung jumlah dokumen pada masing-masing kategori, frekuensi token per kelas, total token per kelas, dan ukuran *vocabulary*.

Selanjutnya, sistem menghitung prior untuk setiap kategori berdasarkan proporsi jumlah dokumen tiap kelas terhadap total data training. Setelah itu, sistem menghitung *likelihood* token terhadap masing-masing kelas menggunakan *Laplace smoothing* agar token yang tidak muncul pada data training tidak menghasilkan probabilitas nol. Hasil perhitungan ini disimpan sebagai model yang akan digunakan untuk memproses surat baru. Sebagai bentuk transparansi, hasil training juga disajikan pada training report dalam bentuk ringkasan data training, distribusi kelas, *prior*, *log prior*, ukuran *vocabulary*, serta token dominan per kategori.

3.4.3 Proses Klasifikasi Surat

Ketika admin memilih sebuah surat untuk diklasifikasikan, aplikasi Laravel mengambil atribut pengirim, perihal surat, dan isi surat, kemudian mengirimkannya ke *service* Flask. Data tersebut diproses menggunakan model Naïve Bayes yang telah dilatih. Untuk setiap token hasil *preprocessing*, sistem menghitung nilai *likelihood* terhadap setiap kelas. Nilai-nilai *likelihood* tersebut kemudian digabungkan dengan *prior* tiap kelas menggunakan bentuk logaritma untuk menghasilkan total *log score*. Setelah total *log score* seluruh kelas diperoleh, sistem membandingkan nilainya dan memilih kelas dengan skor tertinggi sebagai hasil klasifikasi akhir. Hasil ini kemudian dikembalikan ke aplikasi Laravel dan disimpan ke basis data, sehingga status klasifikasi, kategori hasil, skor terbaik, dan detail hasil klasifikasi dapat ditampilkan pada halaman admin.

3.4.4 Penjabaran Hasil Perhitungan

Salah satu keunggulan implementasi algoritma pada penelitian ini adalah kemampuan sistem dalam menampilkan penjabaran hasil klasifikasi. Sistem tidak hanya menunjukkan kategori akhir, tetapi juga menampilkan tahapan perhitungan yang meliputi hasil *preprocessing*, *prior* kelas, *token* yang paling berpengaruh, subtotal log probability pada masing-masing *field*, dan ranking semua kelas berdasarkan skor akhir. Dengan adanya penjabaran ini, admin dapat mengetahui alasan mengapa suatu surat diklasifikasikan ke kategori tertentu. Misalnya, surat yang memuat kata-kata

seperti permohonan, izin, pengajuan, atau fasilitas cenderung memiliki skor lebih tinggi pada kategori permohonan. Sebaliknya, surat yang mengandung kata seperti undangan, rapat, kegiatan, atau kehadiran lebih cenderung masuk ke kategori undangan. Surat yang memuat kata pemberitahuan, informasi, atau disampaikan lebih dekat dengan kategori pemberitahuan, sedangkan surat yang mengandung kata laporan, hasil, atau pertanggungjawaban cenderung diklasifikasikan ke kategori laporan.

Penjabaran ini menunjukkan bahwa model bekerja berdasarkan distribusi token yang dipelajari dari data training, bukan secara acak. Hal ini menjadi nilai tambah penting dalam sistem, karena hasil klasifikasi dapat dipahami dan dijelaskan kembali kepada pengguna.

3.4.5 Kelebihan dan Keterbatasan Implementasi

Implementasi algoritma Naïve Bayes pada penelitian ini memiliki beberapa kelebihan. Pertama, algoritma ini relatif ringan dan sesuai untuk diterapkan pada sistem berbasis web. Kedua, model dapat dilatih ulang dengan mudah ketika data training bertambah. Ketiga, hasil klasifikasi dapat dijelaskan secara rinci kepada admin melalui fitur detail klasifikasi dan *training report*. Namun demikian, implementasi ini juga memiliki keterbatasan. Kinerja model sangat dipengaruhi oleh kualitas data training, distribusi kelas, dan variasi redaksi surat. Apabila salah satu kategori memiliki jumlah data yang jauh lebih sedikit, maka model cenderung kurang stabil dalam mengenali kategori tersebut. Selain itu, walaupun *preprocessing* membantu menyeragamkan token, variasi bahasa administratif yang sangat beragam masih dapat memengaruhi hasil klasifikasi.

3.5 Hasil Pengujian

Pengujian dilakukan untuk memastikan bahwa aplikasi pengarsipan surat masuk dan fitur klasifikasi dapat berjalan dengan baik sesuai rancangan sistem. Selain pengujian fungsi aplikasi, dilakukan pula evaluasi terhadap hasil klasifikasi yang dihasilkan oleh algoritma Naïve Bayes menggunakan confusion matrix.

Berdasarkan evaluasi terhadap 16 dokumen surat masuk, hasil klasifikasi yang dihasilkan sistem menunjukkan nilai akurasi sebesar 75,00%, presisi rata-rata sebesar 63,89%, dan recall rata-rata sebesar 72,22%. Hasil tersebut menunjukkan bahwa fitur klasifikasi pada aplikasi mampu melakukan pengelompokan surat ke dalam empat kategori dengan cukup baik sesuai data yang digunakan pada penelitian. Berdasarkan confusion matrix, kategori permohonan memiliki performa terbaik. Dari total 9 data uji pada kelas permohonan, sebanyak 8 dokumen berhasil diprediksi dengan benar, sedangkan 1 dokumen keliru diprediksi sebagai pemberitahuan. Hal ini terlihat juga dari nilai precision dan recall kelas permohonan yang masing-masing mencapai 88,89%. Hasil tersebut menunjukkan bahwa model cukup baik dalam mengenali pola kata yang berkaitan dengan surat permohonan.

Pada kategori undangan, model menunjukkan performa yang cukup stabil dengan nilai precision dan recall masing-masing sebesar 66,67%. Dari 3 data uji kategori undangan, sebanyak 2 dokumen berhasil diklasifikasikan dengan benar, sedangkan 1 dokumen salah diprediksi sebagai permohonan. Hasil ini menunjukkan bahwa sebagian pola bahasa pada surat undangan masih memiliki kemiripan dengan kategori lain, terutama permohonan. Untuk kategori pemberitahuan, performa model masih relatif lemah. Dari 3 data uji kategori pemberitahuan, hanya 1 dokumen yang berhasil diklasifikasikan dengan benar, sedangkan 1 dokumen salah diprediksi sebagai undangan dan 1 dokumen lainnya salah diprediksi sebagai laporan. Nilai precision pada kategori ini sebesar 50,00%, sedangkan recall hanya sebesar 33,33%. Hasil ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan surat pemberitahuan dari kategori lain yang memiliki redaksi serupa.

Sementara itu, kategori laporan memiliki nilai recall sebesar 100,00%, karena satu-satunya data uji pada kategori ini berhasil diprediksi dengan benar. Namun demikian, nilai precision hanya sebesar 50,00%, karena terdapat satu dokumen dari kategori pemberitahuan yang salah diprediksi sebagai laporan. Kondisi ini menunjukkan bahwa performa pada kategori laporan belum dapat dianggap stabil, terutama karena jumlah data uji pada kelas ini masih sangat terbatas. Secara umum, hasil pengujian menunjukkan bahwa sistem mampu melakukan klasifikasi surat masuk secara otomatis dengan performa yang cukup baik. Kinerja terbaik model terlihat pada kategori yang memiliki jumlah data training lebih banyak, yaitu permohonan. Sebaliknya, kategori dengan jumlah data lebih sedikit atau yang memiliki redaksi surat yang mirip dengan kategori lain cenderung menghasilkan performa yang lebih rendah. Hal ini menunjukkan bahwa distribusi data training sangat berpengaruh terhadap hasil klasifikasi. Dari sisi implementasi sistem, hasil pengujian ini menunjukkan bahwa aplikasi tidak hanya mampu menjalankan proses klasifikasi, tetapi juga mampu menyajikan hasil klasifikasi secara transparan melalui detail preprocessing, training report, skor tiap kelas, dan ranking hasil akhir. Dengan demikian, sistem yang dibangun memberikan manfaat praktis dalam mendukung proses pengarsipan surat masuk di BAUK STMIK Widya Cipta Dharma, meskipun masih terdapat ruang pengembangan terutama pada penambahan data training dan peningkatan kualitas data pada kategori yang jumlahnya lebih sedikit.

4. KESIMPULAN

Penelitian ini berhasil membangun sistem pengarsipan dokumen surat masuk berbasis web dengan menerapkan algoritma Naïve Bayes sebagai fitur klasifikasi otomatis. Sistem dikembangkan menggunakan Laravel dan Livewire sebagai aplikasi utama, serta Flask sebagai service klasifikasi. Sistem mampu membantu proses pengelolaan surat masuk secara lebih terstruktur melalui fitur penyimpanan surat, klasifikasi otomatis, detail preprocessing, dan training report. Berdasarkan hasil pengujian aplikasi dan evaluasi hasil klasifikasi terhadap 16 dokumen surat masuk, sistem memperoleh nilai akurasi sebesar 75,00%, presisi rata-rata sebesar 63,89%, dan recall rata-rata sebesar 72,22%. Hasil tersebut menunjukkan bahwa fitur klasifikasi yang diterapkan pada sistem mampu membantu proses pengelompokan surat masuk ke dalam kategori permohonan, undangan, pemberitahuan, dan laporan dengan cukup baik sesuai data yang digunakan pada penelitian. Secara keseluruhan, sistem yang dibangun dapat membantu meningkatkan efisiensi dan konsistensi pengarsipan surat masuk. Penelitian selanjutnya dapat difokuskan pada penambahan jumlah data training, pemerataan distribusi kelas, serta perbandingan dengan algoritma klasifikasi lain untuk meningkatkan kualitas hasil klasifikasi.

REFERENCES

- [1] R. Yusuf, “PERKEMBANGAN PENGELOLAAN ARSIP DI ERA TEKNOLOGI,” *Acarya Pustaka J. Ilm. Perpust. dan Inf.*, vol. 7, no. 1, 2020, doi: 10.23887/ap.v7i1.25634.
- [2] J. F. Indey and S. Supangat, “IMPLEMENTASI ALGORITMA NAÏVE BAYES DALAM SISTEM PENGARSIPAN SURAT BERBASIS AI DI GPI PAPUA KLASIS MIMIKA PAPUA TENGAH,” *J. Ilm. Inform.*, vol. 12, no. 02, pp. 102–113, Sep. 2024, doi: 10.33884/jif.v12i02.9087.
- [3] C. R. Agustina, “Optimalisasi Pelayanan Administrasi Melalui Tata Kelola Kearsipan Kantor Kapanewon Tempel,” *Soc. J. Pengabd. Masy.*, vol. 1, no. 2, pp. 89–94, Apr. 2022, doi: 10.55824/jpm.v1i2.84.
- [4] E. Aprilia and K. N. L. Chamalinda, “Pendampingan Pengarsipan Berkas sebagai Upaya untuk Meningkatkan Efisiensi dan Akuntabilitas dalam Tata Kelola Administrasi,” *J. Pengabd. Masy.*, vol. 1, no. 1, 2025, [Online]. Available: <https://journal.trunojoyo.ac.id/v3/sokla/article/view/156>
- [5] F. R. Gerung, G. D. P. Maramis, and E. R. S. Moningkey, “Penerapan Algoritma Naive Bayes pada Arsip Surat Kantor Kecamatan Motoling Barat,” *Remik Ris. dan E-Jurnal Manaj. Inform. Komput.*, vol. 9, no. 2, 2025, doi: 10.33395/remik.v9i2.14786.
- [6] L. Tantowi and L. Wijayanti, “PELUANG DAN TANTANGAN PENYIMPANAN CLOUD STORAGE PADA DOKUMEN DIGITAL,” *Shaut Al-Maktabah J. Perpustakaan, Arsip dan Dokumentasi*, vol. 15, no. 1, pp. 118–131, Jun. 2023, doi: 10.37108/shaut.v15i1.803.
- [7] E. Martantoh and N. Yanih, “Implementasi Metode Naïve Bayes Untuk Klasifikasi Karakteristik Kepribadian Siswa Di Sekolah MTS Darussa’adah Menggunakan Php Mysql,” *J. Teknol. Sist. Inf.*, vol. 3, no. 2, pp. 166–175, Sep. 2022, doi: 10.35957/jtsi.v3i2.2896.
- [8] M. Agustriya, M. Ula, and Kurniawati, “Analisis Kinerja Algoritma Klasifikasi Naïve Bayes Menggunakan Genetic Algorithm dan Bagging untuk Data Publik Risiko Transaksi Kartu Kredit,” *J. Sist. dan Teknol. Inf.*, vol. 12, no. 3, p. 584, Jul. 2024, doi: 10.26418/justin.v12i3.80136.
- [9] L. B. Adzy, A. Asriyanik, and A. Pambudi, “ALGORITMA NAÏVE BAYES UNTUK KLASIFIKASI KELAYAKAN PENERIMA BANTUAN IURAN JAMINAN KESEHATAN PEMERINTAH DAERAH KABUPATEN SUKABUMI,” *J. Mnemon.*, vol. 6, no. 1, pp. 1–10, May 2023, doi: 10.36040/mnemonic.v6i1.5714.
- [10] D. A. Nur Faradila, L. Magdalena, and M. Febima, “Penerapan Naïve Bayes Classifier Untuk Klasifikasi Postingan Berita Hoaks Di Instagram Cirebon Saber Hoaks,” *Media J. Inform.*, vol. 16, no. 2, p. 243, Dec. 2024, doi: 10.35194/mji.v16i2.4616.
- [11] H. P. Jelita, M. I. Sa’ad, and Wahyuni, “Penerapan Algoritma Naïve Bayes Dalam Analisis sentiment Masyarakat Terhadap STMIK Widya Cipta Dharma,” *Bull. Inf. Technol.*, vol. 6, no. 2, 2025, doi: 10.47065/bit.v6i2.2029.
- [12] D. R. Maharani, L. T. Ningrum, and L. Utari, “Penerapan Metode Naive Bayes Untuk Pengelolaan Status Retensi Arsip,” *TeknoIS J. Ilm. Teknol. Inf. dan Sains*, vol. 15, no. 1, pp. 23–39, Feb. 2025, doi: 10.36350/jbs.v15i1.279.
- [13] I. Kasim, Yuyun, and S. Sahibu, “KLASIFIKASI SURAT DIGITAL MENGGUNAKAN ALGORITMA MACHINE LEARNING,” *J. IT*, vol. 13, no. 2, pp. 66–71, Aug. 2023, doi: 10.37639/jti.v13i2.350.
- [14] Y. Yulindawati, S. Lailiyah, A. Yusnita, and A. Hafifah, “Rekomendasi Pemilihan Judul Tugas Akhir Menggunakan Metode Naïve Bayes,” *J. Inf. Syst. Manag.*, vol. 5, no. 2, pp. 171–175, 2024, doi: 10.24076/joism.2024v5i2.1383.
- [15] A. Widyantoro, F. Faradisa Al Bina, T. Prayoga, R. Safei, and M. Akmal Arrasid, “Systematic Literature Review: Membandingkan Pendekatan Metode Agile dan Waterfall Dalam Pengembangan Perangkat Lunak,” *J. Compr. Sci.*, vol. 4, no. 1, pp. 183–193, Jan. 2025, doi: 10.59188/jcs.v4i1.2969.
- [16] M. Subli, R. Muslim, Z. Zaenudin, and A. Nadriati, “Sistem Informasi Pengolahan Data Pemesanan Makanan Dan Minuman Berbasis Website Menggunakan Framework Codeigniter,” *Informatics Digit. Expert*, vol. 3, no.



- 2, pp. 38–45, Nov. 2021, doi: 10.36423/index.v3i2.835.
- [17] Nurwati and Y. Santoso, “ANALISIS PERBANDINGAN K-NEAREST NEIGHBORS DAN NAIVE BAYES UNTUK REKOMENDASI PILIHAN PROGRAM STUDI BAGI MAHASISWA,” *IDEALIS Indones. J. Inf. Syst.*, vol. 8, no. 1, 2025, doi: 10.36080/idealism.v8i1.3344.
- [18] S. H. Ramadhani and M. I. Wahyudin, “Analisis Sentimen Terhadap Vaksinasi Astra Zeneca pada Twitter Menggunakan Metode Naïve Bayes dan K-NN,” *J. JTik (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 6, no. 4, pp. 526–534, Feb. 2022, doi: 10.35870/jtik.v6i4.530.
- [19] M. U. Albab, Y. K. P., and M. N. Fawaiq, “Optimization of the Stemming Technique on Text Preprocessing President 3 Periods Topic,” *J. Transform.*, vol. 20, no. 2, pp. 1–12, Jan. 2023, doi: 10.26623/transformatika.v20i2.5374.
- [20] F. A. Ramadhan, S. H. Sitorus, and T. Rismawan, “Penerapan Metode Multinomial Naïve Bayes untuk Klasifikasi Judul Berita Clickbait dengan Term Frequency - Inverse Document Frequency,” *J. Sist. dan Teknol. Inf.*, 2023, doi: 10.26418/justin.v11i1.57452.

