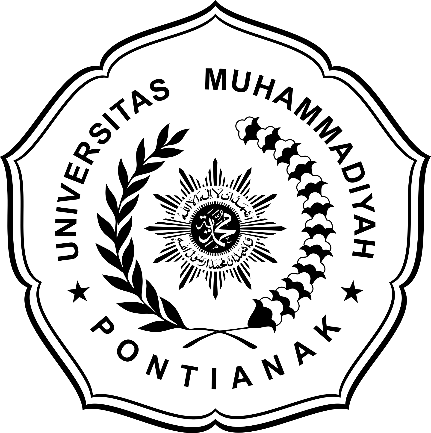
PREDIKSI CUACA DI KABUPATEN SAMBAS MENGGUNAKAN ALGORITMA

RANDOM FOREST

TUGAS AKHIR



OLEH:

|  |
| --- |
| DEDY KURNIAWAN |
| NPM. 191220057 |

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK DAN ILMU KOMPUTER

UNIVERSITAS MUHAMMADIYAH PONTIANAK

2025

# BAB I PENDAHULUAN

## Latar Belakang

Prediksi cuaca merupakan aspek penting dalam berbagai sektor, termasuk pertanian, transportasi, dan manajemen bencana. Di Kabupaten Sambas, yang terletak di Kalimantan Barat, Indonesia, perubahan iklim dan pola cuaca yang tidak menentu menuntut pengembangan model prediksi yang lebih akurat untuk membantu masyarakat dan pemerintah dalam perencanaan dan mitigasi risiko.

Cuaca yang tidak terduga dapat berdampak signifikan pada kegiatan ekonomi dan sosial. Misalnya, petani membutuhkan informasi akurat tentang curah hujan untuk menentukan waktu tanam dan panen. Selain itu, sektor pariwisata juga bergantung pada kondisi cuaca yang baik untuk menarik pengunjung. Oleh karena itu, model prediksi cuaca yang efektif sangat diperlukan untuk memberikan informasi yang tepat waktu dan akurat.

Algoritma Random Forest adalah salah satu teknik machine learning yang telah terbukti efektif dalam berbagai aplikasi prediksi, termasuk prediksi cuaca. Metode ini bekerja dengan membangun banyak pohon keputusan dan menggabungkan hasilnya untuk meningkatkan akurasi prediksi. Keunggulan Random Forest terletak pada kemampuannya untuk menangani data dengan banyak atribut dan mengurangi risiko overfitting dibandingkan dengan model lain.

Dalam konteks penelitian di Kabupaten Sambas, penggunaan algoritma Random Forest dapat dilakukan dengan langkah-langkah berikut:

1. Pengumpulan Data: Mengumpulkan data historis cuaca dari stasiun meteorologi setempat, termasuk suhu, kelembaban, kecepatan angin, dan curah hujan.
2. Pengumpulan Data: Mengumpulkan data historis cuaca dari stasiun meteorologi setempat, termasuk suhu, kelembaban, kecepatan angin, dan curah hujan.
3. Pengumpulan Data: Mengumpulkan data historis cuaca dari stasiun meteorologi setempat, termasuk suhu, kelembaban, kecepatan angin, dan curah hujan.
4. Pengumpulan Data: Mengumpulkan data historis cuaca dari stasiun meteorologi setempat, termasuk suhu, kelembaban, kecepatan angin, dan curah hujan.
5. Pengumpulan Data: Mengumpulkan data historis cuaca dari stasiun meteorologi setempat, termasuk suhu, kelembaban, kecepatan angin, dan curah hujan.

Penelitian sebelumnya menunjukkan bahwa algoritma Random Forest dapat menghasilkan akurasi yang baik dalam memprediksi curah hujan. Sebagai contoh, penelitian di Bandar Lampung menggunakan algoritma ini menunjukkan nilai MAE sekitar 0.111 hingga 0.1295 dan RMSE antara 0.2218 hingga 0.2523[1]. Hasil ini menunjukkan bahwa meskipun Random Forest mungkin tidak selalu seakurat metode lain seperti C4.5 dalam beberapa kasus, ia tetap merupakan pilihan yang solid untuk aplikasi prediksi cuaca.

Dengan menggunakan algoritma Random Forest dalam prediksi cuaca di Kabupaten Sambas, diharapkan dapat meningkatkan akurasi dan keandalan informasi cuaca yang diterima oleh masyarakat. Ini akan membantu dalam perencanaan kegiatan sehari-hari serta mitigasi risiko bencana terkait cuaca ekstrem. Pengembangan lebih lanjut dalam penelitian ini dapat berkontribusi pada sistem informasi cuaca yang lebih efektif di wilayah tersebut.

## Rumusan Masalah

1. Bagaimana kinerja algoritma Random Forest dalam memprediksi kondisi cuaca di Kabupaten Sambas?
2. Faktor-faktor meteorologi apa saja yang paling berpengaruh terhadap keakuratan prediksi cuaca di wilayah tersebut?
3. Seberapa akurat model prediksi cuaca berbasis algoritma Random Forest?

## Batasan Masalah

1. Penelitian ini berfokus pada implementasi data mining menggunakan Algoritma Random Forest untuk prediksi cuaca.
2. Penelitian ini difokuskan pada prediksi cuaca untuk wilayah Kabupaten Sambas saja.
3. Penelitian ini akan menggunakan dataset yang tersedia di website BMKG dengan 11 atribut.
4. Data yang digunakan sebanyak 366 data.
5. Preprocessing data dilakukan dengan menggunakan bahasa pemrograman python dan visual studio code sebagai text editor.

## Tujuan Penelitian

1. Menerapkan algoritma Random Forest untuk memprediksi kondisi cuaca di Kabupaten Sambas.
2. tujuan penelitian ini juga membangun aplikasi praktis dari model prediktif yang dikembangkan, seperti penggunaan perkiraan cuaca untuk keperluan sehari-hari.

## Manfaat Penelitian

1. **Bagi Masyarakat dan Pemerintah Daerah:** Penelitian ini dapat memberikan informasi prediksi cuaca yang lebih akurat, sehingga masyarakat dan pemerintah daerah dapat mengambil langkah preventif dalam menghadapi kondisi cuaca ekstrem yang dapat berdampak pada sektor pertanian, perikanan, dan aktivitas ekonomi lainnya.
2. **Bagi Peneliti dan Akademisi:** Hasil penelitian ini dapat menjadi referensi dan bahan kajian lebih lanjut dalam pengembangan metode prediksi cuaca menggunakan teknik pembelajaran mesin, khususnya algoritma Random Forest.
3. **Bagi Pengembangan Teknologi Prediksi Cuaca**: Model prediksi cuaca berbasis algoritma Random Forest yang dihasilkan dapat menjadi acuan untuk meningkatkan sistem prediksi cuaca yang ada, sehingga lebih adaptif dan akurat dalam menghadapi perubahan iklim yang dinamis.

## Metodologi Penelitian

Penelitian dilakukan untuk memprediksi perkiraan cuaca menggunakan data mining dengan metode Random Forest. Langkah-langkah yang dilakukan dalam penelitian ini diantaranya Idetifikasi Masalah, Studi Literatur, Pengumpulan Data, Implementasi, Pengujian Sistem, dan Pengambilan Kesimpulan.

## Sistematika Penulisan

Struktur penulisan penelitian ini terdiri dari beberapa bab yaitu pendahuluan, landasan teori, metode penelitian, analisis dan perancangan sistem, hasil dan pengujian, serta kesimpulan dan saran.

|  |  |
| --- | --- |
| **BAB I** | **Pendahuluan**  Pada bab pendahuluan ini menjelaskan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penelitian. |
| **BAB II** | **Landasan Teori**  Pada bab ini berisikan tinjauan pustaka mengulas teori-teori terkait Collaborative Filtering serta mengkaji penelitian-penelitian terdahulu |
| **BAB III** | **Metode Penelitian**  Pada bab ini membahas rancangan penelitian, pengumpulan dan analisis data, serta langkah-langkah pengembangan sistem prediksi. |
| **BAB IV** | **Analisis dan Perancangan Sistem**  Pada bab ini membahas analisis kebutuhan sistem berdasarkan hasil pengumpulan dan analisis data. Sistem rekomendasi ini dikembangkan dalam bentuk platform website. |
| **BAB V** | **Hasil dan Pengujian**  Pada bab ini dilakukan untuk menguji kinerja dan fungsionalitas sistem rekomendasi. Selanjutnya di uji kembali apakah metode berhasil sesuai yang diharapkan untuk menarik kesimpulan dan saran. |
| **BAB VI** | **Kesimpulan dan Saran**  Pada bab ini merumuskan kesimpulan dari hasil penelitian dan memberikan saran-saran untuk pengembangan lebih lanjut. |

# BAB II LANDASAN TEORI

## Tinjauan Pustaka

Tinjauan pustaka dilakukan untuk mengarahkan penelitian ini dengan memastikan fokus yang tepat dan menghindari duplikasi dari penelitian sebelumnya. Tinjauan pustaka juga mencakup ulasan tentang penelitian terdahulu yang relevan dengan penelitian yang akan dilakukan, yang akan menjadi acuan dan referensi bagi penulis. Terdapat beberapa penelitian yang membahas mengenai cuaca dan beberapa penelitian yang membahas algoritma Random Forest.

1. Penelitian oleh Zian Asti Dwiyanti, Cahyo Prianto pada tahun 2023 dengan judul “Prediksi Cuaca Kota Jakarta menggunakan Metode Random Forest: Studi Optimalitas”. Tujuan dari penelitian ini adalah untuk membuat prediksi cuaca di Kota Jakarta dengan menggunakan metode Random Forest dan data cuaca historis yang dapat diandalkan dari website OpenData Jakarta. Hasil evaluasi menunjukkan bahwa model Random Forest memberikan prediksi cuaca yang baik dengan akurasi, presisi, dan recall sebesar 0,71, skor F1 sebesar 0,70, dan ROC-AUC sebesar 0,92. [2].
2. Penelitian oleh Ghaitsa Amany Mursianto, Isma’il Muhammad Falih, Muhammad Irfan, Tiara Sakinah, Desta Sandya Prasvita pada tahun 2021 dengan judul “Perbandingan Metode Klasifikasi Random Forest dan XGBoost Serta Implementasi Teknik SMOTE pada Kasus Prediksi Hujan”. Tujuan penelitian ini adalah untuk mengklasifikasikan prediksi hujan pada hari-hari berikutnya dengan menggunakan metode klasifikasi Random Forest. Sistem prediksi cuaca yang telah kami buat mendapatkan klasifikasi Random Forest dengan resampling dengan akurasi 95,59%, tetapi XGBoost mendapatkan akurasi tertinggi pada klasifikasi tanpa resampling dengan akurasi 94,34% [3].
3. Penelitian oleh Agung Hot Iman, Fransisco Ready Permana, Gito Putro Wardana, Raihan Kemmy Rachmansyah, Mayanda Mega Santoni pada tahun 2022 dengan judul “Perbandingan Algoritma Klasifikasi Random Forest dan Extreme Gradient Boosting pada Dataset Cuaca Provinsi DKI Jakarta Tahun 2018”. Setelah melewati tahap preprocessing, analisis data ekstensif, pengecekan label, dan pelatihan data dilakukan untuk mendapatkan hasil yang akurat. Hasil penelitian menunjukkan bahwa, jika digunakan pada data cuaca Provinsi DKI Jakarta tahun 2018, algoritma Random Forest akan lebih baik daripada algoritma Extreme Gradient Boosting. Ini karena waktu pemrosesan yang lebih cepat dan akurasi yang dihasilkan cukup tinggi yaitu 68% [4].
4. Penelitian oleh Risanti, Widyaningrum Indrasari, Haris Suhendar pada tahun 2024 dengan judul “Analisis Model Prediksi Cuaca Menggunakan Support Vector Machine, Gradient Boosting, Random Forest, Dan Decision Tree”. Penelitian ini menggunakan model support vector machine, gradient boosting, random forest, dan decision tree. Perbandingan antara data pelatihan dan tes adalah 70:30. Hasil penelitian menunjukkan bahwa akurasi model support vector machine, gradient boosting, random forest, dan decision tree masing-masing sebesar 0.1697, 0.6696, 0,7918, 0,8416, dan 0,8280. Ini menunjukkan bahwa random forest memiliki pengaruh paling besar terhadap hasil prediksi cuaca dengan hasil akurasi 0.8416 [5].
5. Penelitian oleh M. Imam Whidyarto, Casi Setianingsih, Wendi Harjupa, Risyanto pada tahun 2023 dengan judul “Prediksi Curah Hujan Dari Data Satelit Himawari-8 Menggunakan Metode Random Forest”. Metode Random Forest (RF) menggunakan data netCDF untuk memprediksi ketinggian awan dan hujan. Ini menggunakan titik acuan, yaitu kilometer (km). Akurasi tertinggi adalah 100% dalam pengujian data 120 data. [6].

## Data Mining

Data mining adalah serangkaian prosedur yang digunakan untuk mengidentifikasi pola dan informasi yang berguna dari suatu basis data yang sebelumnya tidak diketahui secara manual. Data mining muncul pada tahun 1990-an dan digunakan secara efektif dan akurat untuk menemukan hubungan antar data dan menemukan pola dan informasi yang berguna untuk pengelompokan data dalam satu atau lebih cluster sehingga objek-objek dalam cluster memiliki tingkat kemiripan yang tinggi satu sama lain. Bagian dari proses penemuan pengetahuan dari basis data, Knowledge Discovery in Databases, adalah data mining. [7].

Menurut Kusrini dan Taufiq Luthfi (2009), pemrosesan data dipilah menjadi beberapa kategori berdasarkan fungsi yang dapat dilakukan, seperti:

1. Deskripsi

Kadang-kadang, analis dan penelitian hanya ingin mencari cara untuk menunjukkan pola dan kecenderungan dalam data.

1. Estimasi

Kecuali variable target estimasi lebih ke arah numerik daripada ke arah kategori, estimasi hampir sama dengan klasifikasi. Model dibangun dengan recordlengkap yang menyediakan nilai dan variable target sebagai nilai prediksi; kemudian, pada peninjauan berikutnya, nilai estimasi variable target didasarkan pada nilai variable prediksi.

1. Prediksi

Prediksi dan klasifikasi hampir sama, kecuali prediksi nilai dari hasil di masa mendatang.

1. Klasifikasi

Dalam klasifikasi, target variable kategori digunakan. Sebagai contoh, penggolongan pendapat dapat dikategorikan menjadi tiga kategori: pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

1. Pengklusteran

Kumpulan catatannya yang mirip satu sama lain dan tidak sama dengan catatannya dalam kluste lain disebut "klusteran". Kluster adalah kumpulan catatannya yang dikelompokkan, diamati, atau diperhatikan untuk membentuk kelas catatannya yang memiliki kemiripan.

1. Asosisasi

Menemukan fitur yang muncul dalam satu waktu adalah tugas bersama dalam datamining. Analisis keranjang belanja lebih dikenal di dunia bisnis [8].

## Random Forest

Seperti namanya, Random forest terdiri dari banyak pohon keputusan individu yang bekerja sama sebagai kelompok. Setiap pohon random forest individu menghasilkan prediksi kelas, dan kelas dengan suara terbanyak dimasukkan ke dalam prediksi model. Visualisasi model Random forest membuat perkiraan. Konsep dasar Random forest sederhana tetapi kuat. [9]. Algoritma pohon random (RF) menggunakan pemisah biner rekursif untuk mencapai node akhir dalam struktur pohon keputusan, yang pada dasarnya terdiri dari pohon regresi dan klasifikasi. Hutan acak menghasilkan banyak pohon independen dengan menggunakan teknik bootstrap dan subset acak dari data pelatihan dan variabel input. Aggregasi bootstrap, juga disebut bagging, melibatkan penggunaan teknik bootstrap atau pengambilan sampel berkali-kali untuk membuat set instruksi yang berbeda dengan menggantikan data instruksi asli. Setiap data pelatihan yang dikumpulkan selama proses ini digunakan untuk membuat pohon klasifikasi. Hasil klasfisikasi adalah agregasi suara terbanyak, dan hasil regresi adalah rata-rata. Berikut adalah struktur umum dari random forest yang ditunjukan Gambar 1 [10].

Langkah-langkah Algoritma Random Forest:

### Persiapan Data

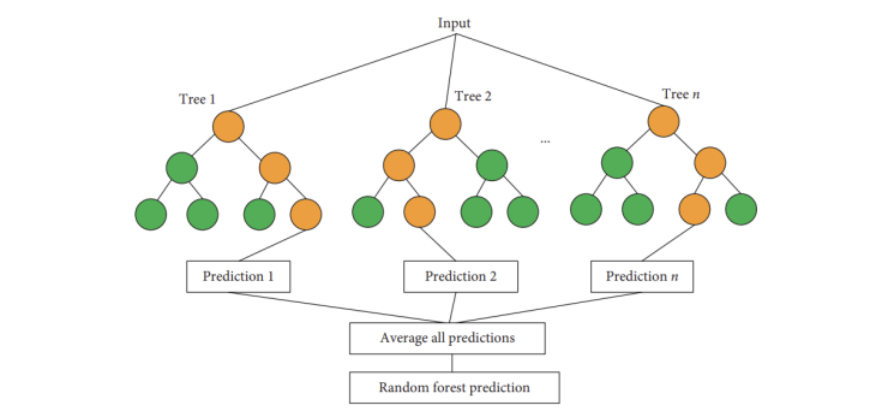
* Kumpulkan dataset yang berisi fitur (variabel independen) dan label (variabel target). Misalnya, untuk prediksi cuaca, fitur dapat mencakup suhu, kelembaban, curah hujan, dll.
* Bagi dataset menjadi data latih (training set) dan data uji (testing set).

### Membangun Pohon Keputusan (Decision Trees)

* Tentukan jumlah pohon keputusan (misalnya, 100 atau 500 pohon) yang akan digunakan dalam model Random Forest.
* Untuk setiap pohon keputusan:
  + Lakukan bootstrap sampling, yaitu memilih sampel secara acak dari data latih dengan penggantian. Dengan demikian, setiap pohon dibangun menggunakan subset data yang berbeda.
  + Pilih secara acak subset fitur dari semua fitur yang tersedia. Ini dilakukan untuk setiap pembagian simpul pada pohon untuk mengurangi korelasi antara pohon.
  + Bangun pohon keputusan dengan menggunakan subset data dan fitur yang dipilih. Setiap pohon tumbuh secara penuh tanpa memangkas cabangnya.

### Melakukan Prediksi

* Untuk membuat prediksi, setiap pohon memberikan prediksi sendiri untuk data uji atau data baru.
* Prediksi akhir untuk klasifikasi ditentukan dengan metode voting mayoritas (prediksi yang paling sering muncul di antara semua pohon).
* Untuk regresi, prediksi akhir ditentukan dengan menghitung rata-rata prediksi dari semua pohon.



Gambar 1 Struktur umum model *Random Forest* [10]

## Visual Studio Code

Visual Studio Code adalah editor teks Microsoft yang sederhana dan kuat yang kompatibel dengan berbagai sistem operasi, termasuk Linux, Mac, dan Windows. Ini mendukung bahasa pemrograman seperti JavaScript, Typescript, dan Node.js secara langsung, dan mendukung bahasa pemrograman lainnya dengan plugin yang dapat diunduh dari marketplace Visual Studio Code, seperti C++, C#, Python, Go, Java, PHP, dan lainnya [11].

## Python

Python adalah bahasa pemrograman bersifat open source yang dioptimalkan untuk kualitas software, produktivitas developer, portabilitas program, dan integrasi komponen. Python saat ini merupakan salah satu dari empat atau lima bahasa pemrograman yang paling populer di dunia dan digunakan untuk membangun berbagai jenis perangkat lunak, seperti skrip internet, pemrograman sistem, antarmuka pengguna, penyesuaian produk, dan penghitungan numerik programming.

Python, sebagai bahasa pemrograman, menawarkan serangkaian fitur yang dapat digunakan oleh para pengembang perangkat lunak. Berikut beberapa dari fitur-fitur yang dimiliki oleh bahasa pemrograman Python::

1. Multi Paradigm Design.
2. Open Source.
3. Simplicity.
4. Library Support.
5. Portability.
6. Extendable.
7. Scalability [12].

# BAB III METODE PENELITIAN

Studi Literatur

Preprocessing Data dan Exploratory

Data Analysis

Pembagian Data

## Data Latih

## Data Uji

Modeling & Tuning

Evaluasi

Kesimpulan

Gambar 2 Tahapan Metode Penelitian

## Studi Literatur

Pada tahapan ini peneliti melakukan studi literatur untuk mendapatkan teori-teori yang valid terkait dengan penelitian yang dilakukan sebelumnya. Sumber literatur yang digunakan untuk dijadikan referensi antara lain jurnal, artikel, dan situs internet.

## Input Dataset

Data yang digunakan untuk penelitian ini diambil dari open dataset BMKG

## Preprocessing Data dan Exploratory Data Analysis

Tahapan selanjutnya, peneliti melakukan pra-proses dan exploratory data analysis pada data yang telah peroleh. Tahapan ini bertujuan untuk membersihkan data dari derau (noise), mendapatkan lebih banyak insight dari data dengan exploratory data analysis, dan mengekstrak informasi yang ada pada data dengan melakukan feature engineering.

## Pembagian Data

Tahapan setelah preprocessing dan exploratory data analysis adalah pembagian data menjadi dua yaitu data latih dan data uji dengan perbandingan 80% data latih dan 20% data uji. Selanjutnya akan dipisahkan antara kolom label dan kolom fitur. Pada kolom label dilakukan pembagian secara stratifikasi atau pembagian data secara seimbang karena label yang digunakan bertipe kategorikal.

## Preprocessing, Modeling, dan Tuning

3.5.1 Preprocessing

Sebelum tahapan selanjutnya, peneliti mempersiapkan data terlebih dahulu agar lebih siap untuk di proses. Peneliti melakukan tahapan pra-proses kembali setelah tahap pembagian data. Hal ini peneliti lakukan untuk menghindari kebocoran data sehingga akan mengurangi tingkat overfitting ataupun underfitting. Pada tahapan pra-proses kali ini, peneliti menggunakan Column Transformer dengan membagi kolom menjadi kolom numerik dan kolom kategorik. Untuk kolom numerik dilakukan pra-proses berupa Polynomial Feature sedangkan kolom kategorik dilakukan pra-proses berupa One-Hot Encoder dengan tujuan untuk membangun model yang sederhana yang dapat digunakan untuk pemahaman dan melakukan prediksi. Polynomial Feature adalah sebuah teknik untuk mengatasi data yang tidak linier dengan menambahkan kekuatan pada setiap fitur sebagai fitur yang baru. Polyomial Feature akan melatih model linier pada kumpulan fitur yang telah diperluas. One-Hot Encoder sendiri adalah salah satu metode encoding yang merepresentasikan data bertipe kategori sebagai vektor biner dengan nilai integer, 0 dan 1, dimana semua elemen akan bernilai 0 kecuali satu elemen yang bernilai 1, yaitu elemen yang memiliki nilai kategori tersebut.

3.5.2 Modeling

Setelah membuat Column Transformer, Kemudian peneliti membuat Pipeline untuk menggabungkan Column Transformer yang telah dibuat sebelumnya dengan algoritma yang akan peneliti gunakan yaitu Random Forest dan Extreme Gradien Boosting. Random Forest adalah pengembangan dari metode pohon keputusan yang menggunakan beberapa pohon keputusan, dimana setiap pohon keputusan telah dilakukan pelatihan menggunakan sampel individu dan setiap atribut dipecah pada pohon yang dipilih antara atribut subset yang bersifat acak. Sementara itu, Extreme Gradient Boosting adalah metode boosting yang menggabungkan beberapa pohon keputusan dan dibangun dengan cara membuat pohon keputusan yang baru berdasarkan residual pohon keputusan sebelumnya dengan tujuan untuk menghasilkan kesalahan prediksi (error) yang lebih kecil dari model sebelumnya. Pipeline sendiri adalah sebuah metode yang dapat mengatur aliran data input dan output dari sebuah model atau kumpulan beberapa model machine learning. Salah satu keuntungan menggunakan pipeline adalah pipeline dapat memproses data mulai dari input data mentah, fitur, output, model machine learning dan parameter model, dan output prediksi.

3.5.3 Tuning

Pembuatan Column Transformer dan pipeline bertujuan untuk mempermudah proses dan memprediksi data baru dengan model telah dibuat serta memungkinkan kita dalam melakukan tuning hyperparameter berdasarkan algoritma yang ada di dalam pipeline. Tuning hyperparameter sendiri merupakan proses untuk menemukan nilai hyperparameter terbaik yang sesuai dengan data yang dimiliki sehingga dapat menghasilkan model dengan performa yang terbaik.

**3.6 Evaluasi**

Setelah semua tahapan selesai, selanjutnya akan dilakukan tahapan evaluasi performa dari model Random Forest dan Extreme Gradient Boosting menggunakan data uji. Evaluasi model bertujuan untuk membuat estimasi generalisasi error dari model yang dipilih, dan menguji seberapa baik kinerja model tersebut pada data baru yaitu data uji. Peneliti menggunakan confusion matrix untuk mengetahui efisiensi kinerja model agar dapat diketahui seberapa banyak data yang salah diklasifikasi pada data uji dan classification report untuk mengukur tingkat akurasi, presisi, dan recall.

# 

# BAB IV ANALISIS DAN PERANCANGAN SISTEM

Bab ini menyajikan pembahasan serta perancangan sistem prediksi cuaca. Pembahasan difokuskan pada penguraian kebutuhan-kebutuhan yang diperlukan dalam proses pengembangan sistem.

## Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan merupakan proses yang dilakukan sebelum mengerjakan tugas akhir ini tentang Prediksi Cuaca Di Kabupaten Sambas Menggunakan Algoritma Random Forest. yang mencakup kebutuhan untuk perangkat keras dan perangkat lunak.

1. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang dibutuhkan dalam mengerjakan penelitian tugas akhir ini agar pengerjaan dapat dilakukan karena dalam proses *machine learning* memerlukan perangkat yang mendukung agar dapat berjalan dengan lancar, pada penelitian ini penulis menggunakan spesifikasi perangkat keras sebagai berikut:

1. Prosesor : Intel(R) Core(TM) i5-5300U CPU @ 2.30GHz (4 CPUs), ~2.3GHz.
2. RAM : 8 GB.
3. System type : Windows 10 Pro 64-bit (10.0, Build 19042).

Sedangkan perangkat keras yang dibutuhkan untuk menggunakan aplikasi ini adalah spesifikasi minimum perangkat keras yaitu bisa membuka *web browser* berupa chrome atau yang lainnya.

1. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan dalam membangun aplikasi adalah sebagai berikut:

1. *Jupyter Notebook*, digunakan untuk menulis baris atau kode program dalam tahapan analisis sentimen.
2. *Visual Studio Code,* digunakan untuk menulis baris atau kode program dalam membangung aplikasi.
3. *Anaconda Navigator*, digunakan untuk mengelola *environment* *python* yang terisolasi.
4. *Google Chrome,* digunakan untuk pengguna berinteraksi dengan sistem yang dibikin.

## Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan penjabaran kebutuhan yang dibutuhkan pada sistem yang akan dibangun. Pada analisis kebutuhan fungsional mencangkupi deskripsi global dari aplikasi seperti yang terdapat pada tabel berikut:

Tabel 4. Analisis Kebutuhan Fungsional

|  |  |  |
| --- | --- | --- |
| **No** | **Kebutuhan** | **Deskripsi** |
| 1. | Aplikasi website. | Aplikasi diharapkan dapat berfungsi dengan semestinya, seperti dapat memasukan input data yang diperlukan dan menampilkan hasil pemodelan menggunakan metode *Random Forest* dari data yangbaru yang di input. |
| 2. | Aplikasi pemodelan | Pemodelan dapat melakukan klasifikasi untuk membuat prediksi cuaca menggunakan metode *Random Forest*. |

## Analisis Kebutuhan Data

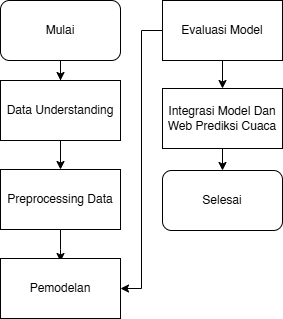
Data yang digunakan merupakan dataset cuaca yang bersumber dari webiste BMKG http://dataonline.bmkg.go.id/home yang telah dilakukan pengunduhan dengan rentang data tahun 2023-2024 dengan jumlah 366 data. Dataset cuaca yang diunduh memiliki fitur-fitur yaitu, antara lain seperti ditunjukan pada Tabel 4.2.

Tabel 4. Fitur dan Keterangan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Fitur** | **Keterangan** | **Jenis Data** |
| 1. | Tanggal | Berisi data waktu, dalam hal ini berformat DD-MM-YYYY | Kategorikal |
| 2. | Tn | Berisi datatemperatur minimum dalam satuan derajat celcius | Numerik |
| 3. | Tx | Berisidata temperatur maksimum dalam satuan derajat celcius | Numerik |
| 4. | Tavg | Berisi datatemperatur rata-rata dalam satuan derajat celcius | Numerik |
| 5. | RH\_avg | Berisi datakelembapan rata-rata dalam satuan derajat celcius | Numerik |
| 6. | RR | Berisidata curah hujan dalam satuan milimeter | Numerik |
| 7. | ss | Berisidata lamanya penyinaran matahari dalam satuan jam | Numerik |
| 8. | ff\_x | Berisidata nilai maksimum kecepatan angin yang tercatat dalam satuan meter per detik | Numerik |
| 9. | ddd\_x | Berisi data arah angin yang tercatat pada saat kecepatan angin mencapai nilai maksimum dalam satuan derajat | Numerik |
| 10. | ff\_avg | Berisi data nilai kecepatan angin rata-rata dalam satuan meter per detik | Numerik |
| 22 | ddd\_car | Berisi data arah angin terbanyak | Kategorikal |

## Perancangan Proses Diagram Aliran

Pada perancangan ini memaparkan tahapan penelitian serta proses pengolahan data guna memastikan hasil yang diperoleh dapat berfungsi secara optimal dalam melakukan prediksi cuaca.

****

Gambar 4. Perancangan Proses Diagram Aliran

Gambar 4.1 menjelaskan alur penelitian dan langkah-langkah mengolah data. Adapun penjelasan dari langkah-langkap pada gambar 4.1 sebagai berikut:

1. *Data Understanding*

Pada tahap ini berisi pengambilan data dan memahami dataset yang digunakan. Pengumpulan data dalam penelitian ini dengan cara *web* *scaping,* denganmenggunakan pustaka *google-play-scrapper*, kemudian data dilanjutkan denan tahapan *labeling.*

1. *Preprocessing*

*Preprocessing* merupakan proses penyiapan data agar data dapat diolah dan proses klasifikasi dapat dilakukan dengan benar. Adapun tahapan dalam *preprocessing* yaitu *cleaning, case folding, tokenizing,* normalisasi, *filtering,* dan *stemming*.

1. Pembobotan Kata Dengan TF-IDF

Tahapan pembobotan kata yaitu proses dalam mengubah nilai dari suatu *term* dan memberikan indikator dari setiap kata sesuai dengan tingkat kepentingan masing-masing kata dengan metode TF-IDF.

1. Pemodelan

Setelah data melewati tahap pembobotan kata*,* data kemudian dilatih untuk bisa digunakan untuk menganalisis sentimen menggunakan metode *SVM* dan *Random Forest.*

1. Evaluasi

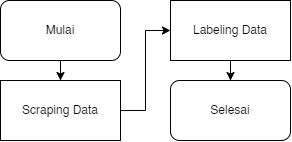
Setelah dilakukan proses pemodelan menggunakan metode *SVM* dan *Random Forest*, maka akan dilakukan evaluasi peforma model menggunakan *confusion matrix*.

1. Integerasi model dan web

Hasil dari pemodelan selanjutnya digunakan untuk dilakukan *integrasi* dari model ke *website.*

### Perancangan *Data Selection*

Pada perancangan ini dijelaskan tahapan *data selection* yang digunakan untuk penelitian ini.



Gambar 4. Perancangan Proses Data Selection

Gambar 4.2 menjelaskan tahapan-tahapan dalam *data selection.* Adapun penjelasan dari tahapan-tahapan pada gambar 4.2 sebagai berikut:

1. *Scraping Data*

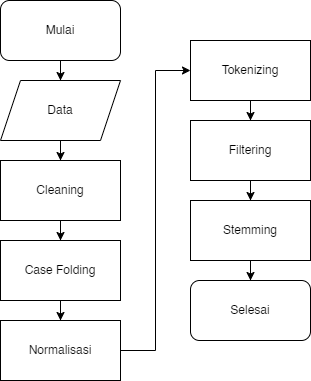
Pengumpulan data dalam penelitian ini dengan cara *web* *scraping,* denganmenggunakan bahasa pemograman *python* serta bantuan pustaka *google-play-scrapper*. Data yang dihasilkan akan disimpan dengan format csv. Penelitian ini memanfaatkan fitur *country=*'id' dari library *google-play-scrapper* supaya pada saat scraping data, hanya ulasan yang berbahasa Indonesia saja yang akan diambil.

1. *Labeling Data*

*Labeling data* merupakan tahapan pemberian nama pada dokumen atau kalimat menjadi kelas positif dan negatif. Dalam penelitian ini menggunakan rating untuk mengetahui kelas positif dan negatif. Dimana nilai dari penamaan sentimen dari rating 1-3 dilabelkan sebagai kelas negatif sedangkan 4-5 dilabelkan sebagai kelas positif*.* Hasil dari palebelan akan disimpan didalam fitur baru bernama *sentiment*. Pada fitur *sentiment* data yang berisi angka 0 berisi sentimen negatif, sedangkan angka 1 berisi sentimen positif.

### Perancangan *Preprocessing*

Pada perancangan ini dijelaskan tahapan preprocessing yang digunakan untuk penelitian ini.



Gambar 4. Perancangan Proses Text Preprocessing

Gambar 4.3 menjelaskan tahapan-tahapan dalam *preprocessing.* Adapun penjelasan dari tahapan-tahapan pada gambar 4.3 sebagai berikut:

1. Data

Data pada tahap ini ialah data yang telah melalui tahapan *data selection*, sehingga data terdiri dari 12 fitur, yaitu *reviewID*, *userName, userImage, content, score, thumbsUpCount, reviewCreatedVersion, at, replyContent, repliedAt, appVersion* dan *sentiment*.

1. *Cleaning*

Tidak semua atribut digunakan pada penelitian ini. Penulis hanya menggunakan atribut *content*, *score* dan *sentiment*, karena pada atribut *content* berisi ulasan pengguna yang akan diklasifikasi, dan atribut *score* serta *sentiment* digunakan untuk pembelajaran model.

1. *Case folding*

Tahapan *case folding* yaitu merubah kalimat menjadi huruf kecil, menghapus angka dari kalimat, menghapus *emoticon* atau symbol pada kalimat dan menghapus tanda baca.

1. Normalisasi

Pada tahapan ini, dilakukan perbaikan penulisan kata yang tidak sesuai dengan kamus KBBI menjadi sesuai.

1. *Tokenizing*

Pada tahapan ini, dilakukan pemotongan kalimat menjadi sebuah kata berdasarkan spasi, dengan bantuan pustaka *nltk*.

1. *Filtering*

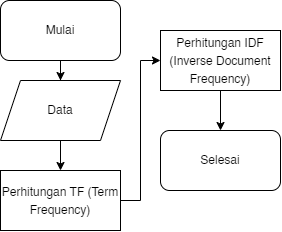
Pada tahapan ini, kata kata yang umum tapi tidak penting akan dihilangkan. Contoh kata-kata tersebut adalah kata depan, kata sambung, kata keterangan dan kata pengganti seperti: “yang”, “ke”, “di”, “sebuah”, “pada”, “oleh”, “ini”, “dari”, dan lain-lain.

1. *Stemming*

Proses *stemming* adalah proses yang dilakukan untuk mendapatkan kata dasar dari sebuah kata, dengan menghilangkan imbuhan atau akhiran pada kata tersebut. *Stemming* dilakukan dengan bantuan pustaka Sastrawi.

### Perancangan Pembobotan Kata Dengan TF-IDF

Pada perancangan ini dijelaskan tahapan pembobotan kata yang digunakan untuk penelitian ini.



Gambar 4. Perancangan Proses Pembobotan Kata

Gambar 4.4 menjelaskan tahapan-tahapan dalam pembobotan kata*.* Adapun penjelasan dari tahapan-tahapan pada gambar 4.4 sebagai berikut:

1. Data

Data pada tahap ini ialah data yang telah melalui tahapan *preprocessing*.

1. PehitunganTF *(Term Frequency)*

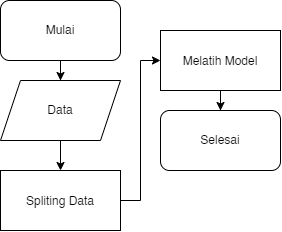
Untuk setiap kata di dalam fitur *content*, hitung berapa kali kata tersebut muncul..

1. PehitunganIDF *(Inverse Document Frequency)*

IDF mengukur seberapa penting suatu kata dalam seluruh fitur *content*. Kata-kata yang sering muncul di fitur *content* umumnya kurang penting daripada kata-kata yang jarang muncul.

### Perancangan Pemodelan

Pada perancangan ini dijelaskan langkah-langkah pada tahapan melatih model.



Gambar 4. Perancangan Pemodelan

Gambar 4.5 menjelaskan tahapan-tahapan dalam melatih model. Adapun penjelasan dari tahapan-tahapan pada gambar 4.5 sebagai berikut:

1. Data

Data pada tahap ini ialah data yang telah melalui tahapan pembobotan kata*.*

1. *Spliting* *Data*

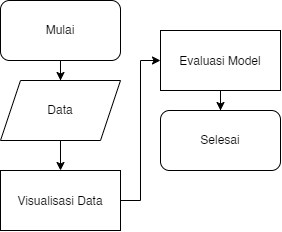
Pada tahapan ini, membagi data *training* dan *testing* dengan4 skenario yaitu dengan perbandingan 90:10, 80:20, 70:30, 60:40*.* Proses ini dilakukan dengan bantuan pustaka sklearn.

1. Melatih model

Pada tahapan ini, data yang sudah terbagi dilakukan implementasi proses klasifikasi sentimen dengan metode *SVM* dan *Random Forest* untuk menentukan hasil klasifikasi sentimen.

### Evaluasi

Pada perancangan ini dijelaskan langkah-langkah pada tahapan evaluasi dari pemodelan.



Gambar 4. Perancangan Pemodelan

Gambar 4.6 menjelaskan tahapan-tahapan dalam melatih model. Adapun penjelasan dari tahapan-tahapan pada gambar 4.6 sebagai berikut:

1. Data

Data pada tahap ini ialah data yang telah melalui tahapan pemodelan*.*

1. *Visualisasi* *Data*

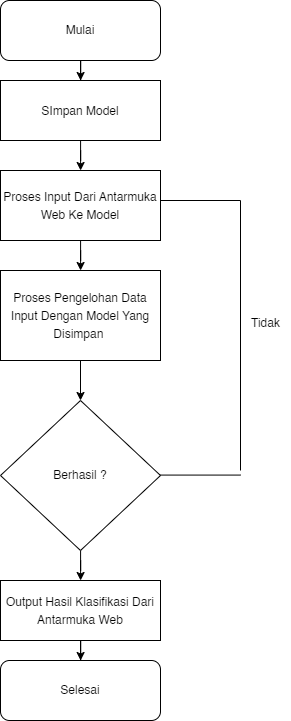
Pada tahapan ini, data yang telah melalui tahapan analisis sentimen akan dipresentasikan ulang dengan harapan mendapatkan informasi baru.

1. Evaluasi Model

Pada tahapan ini, data yang telah dilakukan pemodelan dengan metode *SVM* dan *Random Forest* akan diukur kinerja dari kedua metode. Pengukuran dilakukan menggunakan *confusion matrix* yang akan menghasilkan nilai akurasi, presisi, *recall*, dan *f1-score.* Nilai dari kedua metode ini akan dibandingkan guna mencari metode yang terbaik dalam melakukan analisis sentimen pada ulasan pengguna aplikasi M-Paspor.

### Perancangan *Integrasi* Model dan *Web*

Pada tahapan perancangan ini dijelaskan proses integrasi dari hasil pemodelan yang akan diimplementasikan ke *web* dan dapat melakukan klasifikasi terhadap data text baru.



Gambar 4. Perancangan Proses Integrasi Model

Gambar 4.7 menjelaskan tahapan-tahapan dalam proses *integrasi* model. Adapun penjelasan dari tahapan-tahapan dalam gambar 4.7 sebagai berikut:

1. Simpan model

Menyimpan pemodelan ipynb yang dibangun kedalam file pkl (*Pickle*).

1. Proses *input* dari antarmuka web ke model

Menginput ulasan dari antar muka web kemudian diklasifikasi oleh model.

1. Proses pengolahan data input dengan model yang disimpan

Hasil inputan diproses menggunakan perhitungan permodelan yang disimpan, dalam hal ini akan dilakukan proses klasifikasi dengan metode *SVM* dan *Random Forest.*

1. *Output* hasil klasifikasi dari antarmuka web

*Output* hasil klasifikasi antarmuka web berupa sentimen postitif atau negatif sebuah ulasan yang telah di *input*.

## Perancangan Antarmuka

Syarat aplikasi yang baik adalah memiliki tampilan yang sederhana, *user Friendly,* dimana pengguna (*user*) aplikasi tidak kesulitan dalam memahami cara kerja aplikasi dan mudah menggunakannya.

### Perancangan Antarmuka Halaman Klasifikasi

Pada halaman ini berisikan from *input text* ulasandan pemilihan metode klasifikasiserta output dari klasifikasi sentimen data yang baru berupa text positif atau negatif.

# BAB V HASIL DAN PENGUJIAN

Bab ini berisi tentang hasil, analisis, dan pengujian aplikasi klasifikasi sentimen pada ulasan pengguna aplikasi M-Paspor menggunakan metode *SVM* dan *Random Forest*.

## Hasil Data Selection

Pada tahap ini berisi pengambilan data dan pelabelan data. Pengumpulan data dalam penelitian ini dengan cara *web* *scaping,* denganmenggunakan pustaka *google-play-scrapper*, kemudian data dilanjutkan denan tahapan *labeling*

### Hasil *Scraping Data*

Penulis mendapatkan dataset dengan melakukan *scraping* pada website google play store. Pustaka yang digunakan untuk melakukan *scraping* data adalah google-play-scrapper. Didapatkannya 2.787 data ulasan berbahasa Indonesia dengan 11 macam atribut dari proses *crawling data*.

Tabel 5. Hasil Scraping Data

|  |  |
| --- | --- |
| **Nama Atribut** | **Keterangan** |
| reviewId | Berisi no Id review |
| username | Berisi username pengguna aplikasi M-Paspor |
| userImage | Berisi link gambar profil pengguna |
| content | Berisi komentar pengguna terkait aplikasi M-Paspor |
| score | Berisi angka penilaian yang diberikan untuk aplikasi M-Paspor |
| thumbsUpCount | Berisi jumlah penyuka komentar |
| reviewCreatedVersion | Berisi seri atau versi aplikasi yang  digunakan oleh pengguna |
| at | Berisi waktu pengguna mengunggah  komentar |
| replyContent | Berisi balasan komentar antar pengguna |
| repliedAt | Berisi waktu balasan komentar antar  pengguna |
| appVersion | Berisi seri atau versi aplikasi yang  digunakan oleh pengguna |

Berikut adalah kode program yang digunakan pada tahap *scraping* *data*:

from google\_play\_scraper import Sort, reviews

result, continuation\_token = reviews(

'id.go.imigrasi.paspor\_online',

lang='id',

country='id',

sort=Sort.MOST\_RELEVANT,

count=2787, # defaults to 100

filter\_score\_with=None

**)**

data = pd.DataFrame(np.array(result), columns=['review'])

data = data.join(pd.DataFrame(data.pop('review').tolist()))

### Hasil Pelabelan Data

Pelabelan kelas sentimen melakukan pemberian nama pada dokumen atau kalimat menjadi kelas positif dan negatif. Dalam penelitian ini menggunakan nilai *score* untuk mengetahui kelas positif dan negatif. Dimana nilai dari penamaan sentimen dari nilai *score* 1-3 dilabelkan sebagai kelas negatif sedangkan 4-5 dilabelkan sebagai kelas positif. Berikut kode program yang digunakan pada tahap *labeling data*:

sentimen = []

for index, row in df.iterrows():

if row['score'] <= 3 :

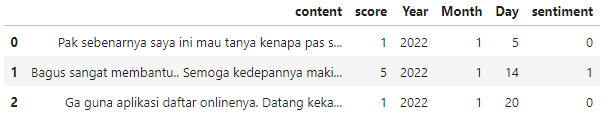
sentimen.append(0)

else:

sentimen.append(1)

df['sentiment'] = sentimen

df.head()



Gambar 5. Hasil Labeling

Gambar 5.1 merupakan hasil dari proses *labeling,* sentiment negative diberi nilai 0 sedangkan sentimen positif diberi nilai 1*.*

## Hasil *Preprocessing*

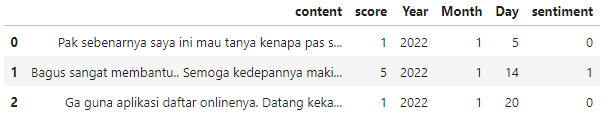
Pada tahap ini akan dilakukan tahapan *preprocessing, preprocessing* adalah proses untuk mempersiapkan data sebelum sebelum dilatih oleh model.

### *Data*

Berikut kode program yang digunakan untuk menyiapkandata hasil *labeling*:

data = pd.read\_csv('data\_mpaspor\_afterLabeling.csv') #loadCsv

df = data.copy() #backup df



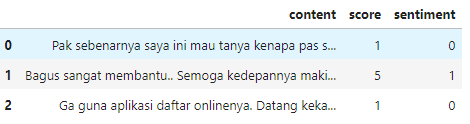
Gambar 5. Hasil Data

Gambar 5.2 merupakan data yang digunakan dalam proses preprocessing, berisis fitur *content, score, Year, Month, Day, sentiment.*

### *Cleaning*

Berikut kode program yang digunakan pada tahap *cleaning data*:

df = df[['content', 'score', 'sentiment']]



Gambar 5. Hasil Cleaning

Gambar 5.3 merupakan data yang telah melalui tahapan *cleaning,* pada tahap ini menghapus fitur yang tidak digunakan pada tahap selanjutnya*.*

### *Case Folding*

*Case folding* adalah tahapan dalam *text preprocessing* yang berfungsi untuk merubah kalimat menjadi huruf kecil, menghapus url, menghapus angka, serta menghapus tanda baca. Berikut kode program yang digunakan pada tahap *case folding*:

df['content'] = df['content'].str.lower() #Merubah Menjadi Lower

df['content'] = df['content'].str.replace("[^\w\s]", ' ', case=False) #Menghapus Tanda Baca

df['content'] = df['content'].str.replace("\s(2)", ' ', case=False) #Menghapus extra whitespace

df['content'] = df['content'].str.replace('https\S+', ' ', case=False) #Menghapus URL dari kolom konten

Tabel 5. Hasil Case Folding

|  |  |
| --- | --- |
| Raw Data | Case Folding |
| Tolong perbaiki sistem nya. Gampang eror masih perlu di kaji lagi biar lebih normal dan tdk menyulitkan pendaftaran | tolong perbaiki sistem nya gampang eror masih perlu di kaji lagi biar lebih normal dan tdk menyulitkan pendaftaran |

### Normalisasi

Pada tahapan ini, dilakukan perbaikan penulisan kata yang tidak sesuai dengan kamus KBBI menjadi sesuai. Berikut kode program yang digunakan dalam tahap normalisasi:

def koreksi\_penulisan(content):

import re

dict\_koreksi = {}

file = open("list koreksi penulisan (tambahan sendiri).txt")

for x in file:

f = x.split(":")

dict\_koreksi.update({f[0].strip(): f[1].strip()})

for awal, pengganti in dict\_koreksi.items():

#content = str(content).replace(awal, pengganti)

content = re.sub(r"\b" + awal + r"\b", pengganti, content)

return content

from datetime import datetime

start\_time = datetime.now()

df["content"] = df["content"].apply(koreksi\_penulisan)

end\_time = datetime.now()

print("Durasi Koreksi Penulisan: {}".format(end\_time - start\_time))

Tabel 5. Hasil Normalisasi

|  |  |
| --- | --- |
| Raw Data | Normalisasi |
| Tolong perbaiki sistem nya Gampang eror masih perlu di kaji lagi biar lebih normal dan tdk menyulitkan pendaftaran | Tolong perbaiki sistem nya Gampang error masih perlu di kaji lagi biar lebih normal dan tidak menyulitkan pendaftaran |

### *Tokenizing*

*Tokenizing* adalah tahapan dalam *preprocessing* yang berfungsi untuk memotong kalimat menjadi sebuah kata berdasarkan spasi. Berikut kode program yang digunakan dalam tahap *tokenizing*:

from nltk.tokenize import word\_tokenize

from nltk.tokenize import RegexpTokenizer

regexp = RegexpTokenizer('\w+')

df['content\_token']=df['content'].apply(regexp.tokenize)

Tabel 5. Hasil Tokenizing

|  |  |
| --- | --- |
| **Input** | **Output** |
| tolong bantu validasi kalau misalnya sudah diterima | ['tolong', 'bantu', 'validasi', 'kalau', 'misalnya', 'sudah', 'diterima'] |
| aplikasinya perlu terus ditingkatkan | ['aplikasinya', 'perlu', 'terus', 'ditingkatkan'] |

### *Filtering*

*Filtering* adalah tahapan dalam *text preprocessing* yang berfungsi untuk menghilangkan kata-kata yang tidak unik, kata kata umum tetapi tidak penting. Contoh kata-kata tersebut adalah kata depan, kata sambung, kata keterangan, serta kata pengganti seperti: ”yang”, ”ke”, “di”, “sebuah”, “pada”, “oleh”, “ini”, “dari”, dan lain-lain. Berikut kode program yang digunakan dalam tahap *filtering* (*stopword removal*):

import nltk

nltk.download('stopwords')

from nltk.corpus import stopwords

# Make a list of indonesian stopwords

stopwords = nltk.corpus.stopwords.words("indonesian")

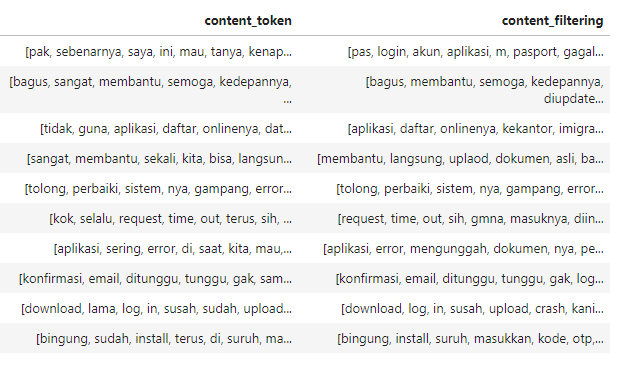
# Extend the list with your own custom stopwords

my\_stopwords = ['mpaspor', 'lol']

stopwords.extend(my\_stopwords)

# Remove stopwords

df['content\_filtering'] = df['content\_token'].apply(lambda x: [item for item in x if item not in stopwords])



Gambar 5. Hasil Filtering (Stopword Removal)

Gambar 5.4 merupakan hasil dari proses *filtering (stopword removal*), *content token* adalah data awal sedangkan *content filtering* adalah data yang telah dilakukan proses *Filtering* (*Stopword Removal*).

### *Stemming*

*Stemming* adalah tahapan dari proses *word normalization* yang berfungsi untuk mendapatkan kata dasar dari sebuah kata, dengan menghilangkan imbuhan atau akhiran pada kata tersebut. Berikut kode program yang digunakan dalam tahap *stemming*:

# import Sastrawi package

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# create stemmer

factory = StemmerFactory()

stemmer = factory.create\_stemmer()

df['stemmed'] = df['content\_filtering'].apply(lambda x: [stemmer.stem(y) for y in x]) # Stem every word.

Berikut merupakan contoh dari proses *tokenizing*:

Tabel 5. Stemming

|  |  |
| --- | --- |
| **Input** | **Output** |
| [‘dipakai’, ‘langkah’, ‘error’, ‘aja’, ‘mekanisme’] | [‘pakai’, ‘langkah’, ‘error’, ‘aja’, ‘mekanisme’] |
| [‘pendaftaran’, ’akun’, ‘email’, ‘typo’] | [‘daftar, ’akun’, ‘email’, ‘typo’] |

## Hasil Pembobotan Kata dengan TF-IDF

Term frequency-inverse document frequency adalah text vectorizer yang mengubah teks menjadi vektor yang dapat digunakan. Ini menggabungkan 2 konsep, Term Frequency (TF) dan Document Frequency (DF).Berikut kode program yang digunakan dalam tahap TF:

term\_frequency\_all = term\_fit.transform(tf)

print(term\_frequency\_all)

Berikut kode program yang digunakan dalam tahap IDF:

X = df['text\_string']

y = df['sentiment']

tfid = TfidfVectorizer()

X\_final = tfid.fit\_transform(X)

## Hasil Pemodelan

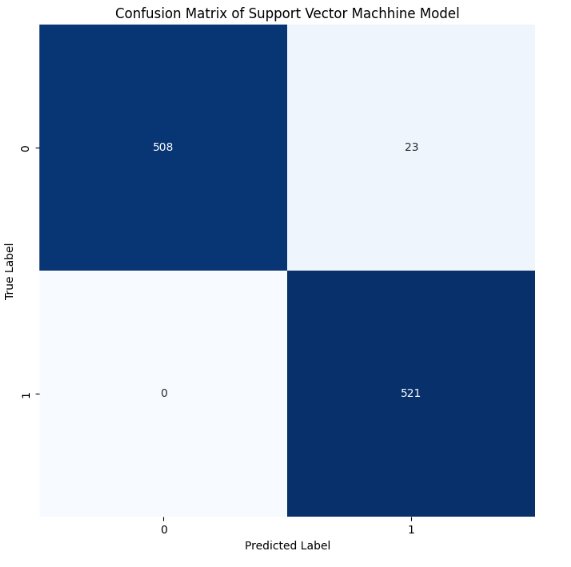
langkah selanjutnya yaitu pemodelan metode *SVM* dan *Random Forest*. Pemodelan dilakukan dengan skenario dataset 80% untuk data latih dan 20% untuk data uji. Berikut adalah hasil pemodelan *SVM* dan *Random Forest* dengan menggunakan skenario data yang telah ditentukan:

Tabel 5. Hasil Akurasi Pemodelan

|  |  |
| --- | --- |
| ***SVM*** | ***Random Forest*** |
| 97.81% | 98.66 |

## Hasil Evaluasi

Evaluasi model menggunakan metode *confusion matrix* untuk mendapatkan akurasi yang dihasilkan oleh model *text mining*. Akurasi digunakan untuk mengidentifikasi keakuratan.



Gambar 5. Hasil Evaluasi SVM

Gambar 5.5 menjelaskan hasil dari *confusion matrix model SVM.* Adapuun rinciannya sebagai berikut:

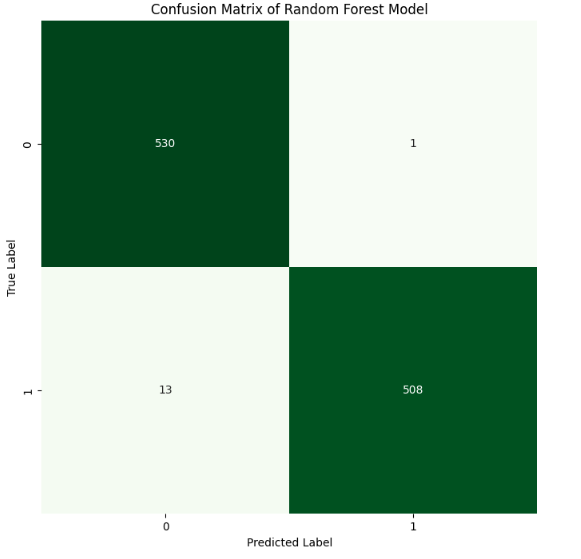
1. *True Positive* (TP), merupakan data *positif* yang diprediksi benar.
2. *True Negative* (TN), merupakan data *negatif* yang diprediksi benar.
3. *False Positive* (FP), merupakan data *negatif* namun diprediksi sebagai data *positif.*
4. *False Negative* (FN), merupakan data *positif* namun diprediksi sebagai data *negative.* Contohnya, berita hoax (*class* 1) tetapi dari model yang dibuat memprediksi berita tersebut non-hoax (*class* 2).

Berikut hasil perhitungan manual dari *confusion matrix*:

Rumus *accuracy =* (TP + TN) / (TP + FP + FN + TN)

Rumus *recall* = TP / (TP + FN)

Rumus *precision* = TP / (TP + FP)



Gambar 5. Confusion Matrix SVM

Gambar 5.6 menjelaskan hasil dari *confusion matrix model Random Forest*

Berikut hasil perhitungan manual dari *confusion matrix model Random Forest*:

Rumus *accuracy =* (TP + TN) / (TP + FP + FN + TN)

Rumus *recall* = TP / (TP + FN)

Rumus *precision* = TP / (TP + FP)

Tabel 5. Hasil Evaluasi

|  |  |  |
| --- | --- | --- |
| **Matrik Evaluasi** | ***SVM*** | ***Random Forest*** |
| Akurasi | 97,81% | 98,66% |
| Presisi | 95,66% | 99,81% |
| Recall | 100% | 97,60% |

## Hasil *Integrasi* Model dan Web

Hasil dari pemodelan selanjutnya digunakan untuk membuat *integrasi* dari model ke *website*, pada fase ini ditampilkan juga hasil dari pembangunan *website*.

### Hasil *Integrasi*

Pada fase ini dijelaskan tahap *integrasi* cara menyimpan dan menggunakan model yang disimpan, pertama adalah menyimpan pemodelan ipynb yang dibangun kedalam file pkl (*Pickle*) menggunakan *library* Python dengan kode program dibawah:

import pickle

with open("model\_rf.pkl", "wb") as f:

pickle.dump(random\_forest\_classifier, f)

Kode program di atas berfungsi untuk menyimpan pemodelan *Random Forest* kedalam file *pickle.*

import pickle

with open("model\_svm.pkl", "wb") as f:

pickle.dump(svm, f)

Kode program di atas berfungsi untuk menyimpan pemodelan *SVM* kedalam file *pickle.* Setelah berhasil menyimpan pemodelan dalam file pkl langkah selanjutnya yaitu dengan memanggil file pkl tersebut kedalam aplikasi *website* yang dibangun menggunakan *library* Python yaitu *flask* kedalam file py seperti pada kode program dibawah:

svc = pickle.load(open("model\_svc.pkl", "rb"))

Kode program di atas berfungsi untuk memanggil file *pickle* kedalam *website* menggunakan *flask*.

### Hasil Halaman Branda

Pada halaman ini ditampilkan hasil dari pembangunan *website* yaitu beranda yang berisi beberapa informasi tentang apa itu berita hoax.



Gambar 5. Halaman Beranda

# BAB VI KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran atau rekomendasi untuk perbaikan, pengembangan, kesempurnaan atau kelengkapan penelitian yang dilakukan.

## Kesimpulan

Berdasarkan hasil analisis sentimen ulasan aplikasi M-Paspor didapatkan bahwa klasifikasi menggunakan metode Random Forest menghasilkan prediksi yang lebih baik dibandingkan dengan metode *SVM*. Hal tersebut ditunjukan dengan nilai akurasi, dan F1-score yang lebih tinggi.

## Saran

Adapun saran penulis untuk penelitian selanjutnya yaitu dapat menerapkan metode lain, seperti Naïve Bayes dan Regresi Logistik atau yang lainya.

# DAFTAR PUSTAKA

[1] Sudarta, *PRAKIRAAN CUACA KOTA BANDAR LAMPUNG DENGAN MENGGUNAKAN RANDOM FOREST DAN C4.5*, vol. 16, no. 1. 2022.

[2] Z. A. Dwiyanti and C. Prianto, “Prediksi Cuaca Kota Jakarta Menggunakan Metode Random Forest,” *J. Tekno Insentif*, vol. 17, no. 2, pp. 127–137, 2023, doi: 10.36787/jti.v17i2.1136.

[3] G. A. Mursianto, I. M. Falih, M. Irfan, T. Sakinah, and D. S. Prasvita, “Perbandingan Metode Klasifikasi Random Forest dan XGBoost Serta Implementasi Teknik SMOTE pada Kasus Prediksi Hujan,” *J. Senamika*, vol. 2, no. 2, pp. 41–50, 2021.

[4] A. Hot Iman, F. Ready Permana, G. Putro Wardana, R. Kemmy Rachmansyah, and M. Mega Santoni, “Perbandingan Algoritma Klasifikasi Random Forest dan Extreme Gradient Boosting pada Dataset Cuaca Provinsi DKI Jakarta Tahun 2018,” *SeminarNasionalMahasiswaIlmuKomputerdanAplikasinya(SENAMIKA)*, pp. 593–601, 2022, [Online]. Available: https://katalog.data.go.id/dataset/data-prakiraan-cuaca-wilayah-provinsi-dki-jakarta-tahun-2018.

[5] R. Risanti, “Analisis Model Prediksi Cuaca Menggunakan Support Vector Machine, Gradient Boosting, Random Forest, Dan Decision Tree,” vol. XII, pp. 119–128, 2024, doi: 10.21009/03.1201.fa18.

[6] H. Nisya, C. Setianingsih, and W. Harjupa, “Prediksi Curah Hujan Dari Data SatelitHimawari-8 Menggunakan Metode K-Nearest Neighbor (KNN),” *e-Proceeding Eng.*, vol. 10, no. 1, pp. 729–735, 2023, [Online]. Available: https://sharaku.eorc.jaxa.jp/GSMaP/.

[7] P. Alkhairi and A. P. Windarto, “Penerapan K-Means Cluster Pada Daerah Potensi Pertanian Karet Produktif di Sumatera Utara,” pp. 762–767, 2019.

[8] A. F. Budiantara and C. Budihartanti, “IMPLEMENTASI DATA MINING DALAM MANAJEMEN INVENTORY PADA PT . MASTERSYSTEM INFOTAMA MENGGUNAKAN METODE ALGORITMA APRIORI,” vol. 7, no. 1, 2020.

[9] A. M. Siregar, “Klasifikasi Untuk Prediksi Cuaca Menggunakan Esemble Learning,” *Petir*, vol. 13, no. 2, pp. 138–147, 2020, doi: 10.33322/petir.v13i2.998.

[10] Ashari Rakhmat Galih and Mutohar Wisnu., “Prakiraan Hujan menggunakan Metode RandomForest dan Cross Validation,” *J. MIND J. | ISSN*, vol. 8, no. 2, pp. 173–187, 2023, [Online]. Available: https://doi.org/10.26760/mindjournal.v8i2.173-187.

[11] K. S. Ningsih, N. J. Aruan, and A. T. A. A. Siahaan, “Aplikasi Buku Tamu Menggunakan Fitur Kamera Dan Ajax Berbasis Website Pada Kantor Dispora Kota Medan,” *SITek J. Sains, Inform. dan Tekonologi*, vol. 1, pp. 94–99, 2022.

[12] H. A. Fauzi, K. A. Putra, and A. Tri, “Analisis Perbandingan Performa Web Service Menggunakan Bahasa Pemrograman Python , Php ,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 237–245, 2018, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j- ptiik/article/view/781.

# DAFTAR GAMBAR

[Gambar 1 Struktur umum model *Random Forest* 10](#_Toc179760948)

[Gambar 2 Tahapan Metode Penelitian 12](file:///D:\AASkripsi\Proposal%20Dedy%20Kurniawan.docx#_Toc179760949)