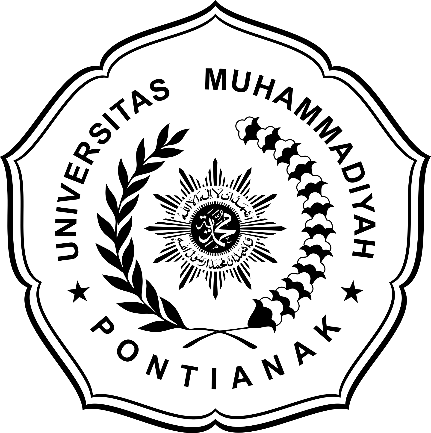
**PERBANDINGAN METODE *SVM* DAN *RANDOM FOREST* TERHADAP ANALISIS SENTIMEN ULASAN**

**APLIKASI M-PASPOR**

TUGAS AKHIR



OLEH:

|  |
| --- |
| M. AFIF TAQIYUDDIN |
| NPM. 171220584 |

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK DAN ILMU KOMPUTER

UNIVERSITAS MUHAMMADIYAH PONTIANAK

2024

# PERNYATAAN KEASLIAN

PERBANDINGAN METODE SVM DAN RANDOM FOREST TERHADAP ANALISIS SENTIMEN ULASAN

APLIKASI M-PASPOR

TUGAS AKHIR

Saya menyatakan bahwa Tugas Akhir ini merupakan tulisan hasil kerja saya sendiri dan bukan orang lain, kecuali kutipan dan ringkasan yang sudah dicantumkan sumbernya.

Pontianak, 10 Juli 2024

Materai 10.000

M. Afif Taqiyuddin

NIM. 171220584

# LEMBAR PERSETUJUAN

Yang bertanda tangan dibawah ini, Dosen Pembimbing Tugas Akhir, menerangkan bahwa:

|  |  |  |
| --- | --- | --- |
| Nama | : | Muhammad Afif Taqiyuddin |
| NIM | : | 171220584 |
| Judul | : | Perbandingan Metode Svm Dan Random Forest Terhadap Analisis Sentimen Ulasan Aplikasi M-Paspor |

DIPERIKSA DAN DISETUJUI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dosen Pembimbing I  Asrul Abdullah, S.Kom., M.Cs |  | Dosen Pembimbing II  Sucipto, M.Kom | | |
| NIDN. 000435345 |  | | NIDN. |

|  |
| --- |
| Mengetahui Dekan Fakultas Teknik dan Ilmu Komputer  Fuazen, S.T., M.T |
| NIDN. 1122087301 |

# LEMBAR PENGESAHAN

Tugas akhir ini telah disidangkan dan dipertahankan di depan tim penguji pada hari Rabu, tanggal 10 bulan Juli tahun 2024 dan diterima sebagai salah satu syarat akhir studi pada Program Studi Teknik Informatika Fakultas Teknik dan Ilmu Komputer Universitas Muhammadiyah Pontianak.

TIM PEMBIMBING

|  |  |  |
| --- | --- | --- |
| Dosen Pembimbing I  Asrul Abdullah, S.Kom., M.Cs |  | Dosen Pembimbing II  Sucipto, M.Kom |
| NIDN. 1128059002 |  | NIDN. 1130038301 |

TIM PENGUJI

|  |  |  |
| --- | --- | --- |
| Dosen Penguji I  Barry Caesar Octariandi, S.Kom., M.Cs |  | Dosen Penguji II  Syarifah Putri Agustini, S.T., M.Kom |
| NIDN. 1125108601 |  | NIDN. 1111088803 |

|  |
| --- |
| Mengetahui  Dekan Fakultas Teknik dan Ilmu Komputer  Fuazen, S.T., M.T |
| NIDN. 1122087301 |

# ABSTRAK

Pemanfaatan dan optimalisasi teknologi digital telah melahirkan bentuk baru birokrasi pemerintahan yaitu *Electronic Government* (*EGovernment*). Perwujudan e-government salah satu nya ialah aplikasi M-Paspor. Aplikasi M-Paspor memiliki banyak pengguna, sehingga banyak pengguna yang memberikan komentar, saran, dan kritikan dalam fitur ulasan di laman Google Play Store. Tujuan penelitian ini untuk mengklasifikasi sentimen pengguna M-Paspor yang terdapat dalam ulasan. Data ulasan pengguna aplikasi M-Paspor yang didapatkan melalui website Google Play dengan jumlah sampel sebanyak 2.787 data ulasan. Ulasan pengguna diberikan label sentimen positif dan negatif. Ulasan tersebut dianalisis menggunakan metode SVM dan Random Forest. Hasilnya, metode Random Forest menghasilkan prediksi yang lebih daripada metode SVM dengan dengan nilai akurasi sebesar 98.66%.

**Kata Kunci:** Data Mining, Machine Learning, Support Vector Machine, Analisis Sentimen, Random Forest

# KATA PENGANTAR

Alhamdulillah segala puji dan syukur kepada Allah SWT, penulis dapat menyelesaikan Skripsi ini yang berjudul “PERBANDINGAN METODE SVM DAN RANDOM FOREST TERHADAP ANALISIS SENTIMEN ULASAN APLIKASI M-PASPOR” atas motivasi yang telah diberikan kepada penulis, oleh karena itu penulis banyak mengucapkan terima kasih yang sebesar besarnya kepada:

1. Kepada Orang Tua tercinta yang menjadi dasar motivasi dalam menyelesaikan skipsi ini. Banyak sekali dukungan yang telah diberikan kepada penulis baik secara moril maupun materi.
2. Bapak Asrul Abdullah, S.Kom., M.Cs. sebagai pembimbing utama yang telah memberikan bimbingan dan arahan yang sangat berguna dalam menyelesaikan skripsi ini.
3. Bapak Sucipto, M.kom. selaku pembimbing kedua dan memberikan bimbingan serta semangat kepada penulis dalam menyelesaikan skripsi ini.
4. Seluruh Dosen dan tenaga Dosen yang pernah mengajar di Program Studi Teknik Informatika Fakultas Teknik yang sudah memberikan ilmu dari awal perkuliahan hingga sekarang.

Saya menyadari bahwa penyusunan Skripsi ini, tidak luput dari kesalahan dan kekurangan. Oleh karena ini, sangat diharapkan saran dan kritik yang konstruktif dari semua pembaca demi kesempurnaan Skripsi ini. Semoga skripsi ini bermanfaat bagi universitas dan pengembangan teknologi informasi dimasa depan.

Pontianak, 10 Juli 2024

M.Afif Taqiyuddin

NPM. 171220584

# DAFTAR ISI

[PERNYATAAN KEASLIAN i](#_Toc171978429)

[LEMBAR PERSETUJUAN ii](#_Toc171978430)

[LEMBAR PENGESAHAN iii](#_Toc171978431)

[ABSTRAK iv](#_Toc171978432)

[KATA PENGANTAR v](#_Toc171978433)

[DAFTAR ISI vi](#_Toc171978434)

[DAFTAR GAMBAR viii](#_Toc171978435)

[DAFTAR TABEL ix](#_Toc171978436)

[1 BAB I PENDAHULUAN 1](#_Toc171978437)

[1.1 Latar Belakang 1](#_Toc171978438)

[1.2 Rumusan Masalah 2](#_Toc171978439)

[1.3 Batasan Masalah 2](#_Toc171978440)

[1.4 Tujuan Penelitian 3](#_Toc171978441)

[1.5 Manfaat Penelitian 3](#_Toc171978442)

[1.6 Metodologi Penelitian 3](#_Toc171978443)

[1.7 Sistematika Penulisan 4](#_Toc171978444)

[2 BAB II LANDASAN TEORI 6](#_Toc171978445)

[2.1 Tinjauan Pustaka 6](#_Toc171978446)

[2.2 M-Paspor 8](#_Toc171978447)

[2.3 Analisis Sentimen 8](#_Toc171978448)

[2.4 Ulasan 8](#_Toc171978449)

[*2.5* *Support Vector Machine (SVM)* 9](#_Toc171978450)

[2.6 *Random Forest* 10](#_Toc171978451)

[*2.7* *Preprocessing* 12](#_Toc171978452)

[2.8 Jupyter Notebook 12](#_Toc171978453)

[2.9 Text Mining 12](#_Toc171978454)

[2.10 *Web Scraping* 12](#_Toc171978455)

[*2.11* *Confusion Matrix* 13](#_Toc171978456)

[3 BAB III METODE PENELITIAN 14](#_Toc171978457)

[*3.1* *Data Selection* 15](#_Toc171978458)

[*3.2* *Preprocessing* 15](#_Toc171978459)

[*3.3* *Transformation* 15](#_Toc171978460)

[*3.4* *Modeling* 16](#_Toc171978461)

[3.5 Evaluasi 16](#_Toc171978462)

[4 BAB IV ANALISIS DAN PERANCANGAN SISTEM 17](#_Toc171978463)

[4.1 Analisis Kebutuhan Non-Fungsional 17](#_Toc171978464)

[4.2 Analisis Kebutuhan Fungsional 18](#_Toc171978465)

[4.3 Analisis Kebutuhan Data 18](#_Toc171978466)

[4.4 Perancangan Proses Diagram Aliran 19](#_Toc171978467)

[*4.4.1* Perancangan *Data Selection* 20](#_Toc171978468)

[*4.4.2* Perancangan *Preprocessing* 21](#_Toc171978469)

[4.4.3 Perancangan Pembobotan Kata Dengan TF-IDF 23](#_Toc171978470)

[4.4.4 Perancangan Pemodelan 23](#_Toc171978471)

[4.4.5 Evaluasi 24](#_Toc171978472)

[4.4.6 Perancangan *Integrasi* Model dan *Web* 25](#_Toc171978473)

[4.5 Perancangan Antarmuka 27](#_Toc171978474)

[4.5.1 Perancangan Antarmuka Halaman Klasifikasi 27](#_Toc171978475)

[5 BAB V HASIL DAN PENGUJIAN 28](#_Toc171978476)

[5.1 Hasil Data Selection 28](#_Toc171978477)

[5.1.1 Hasil *Scraping Data* 28](#_Toc171978478)

[5.1.2 Hasil Pelabelan Data 29](#_Toc171978479)

[5.2 Hasil *Preprocessing* 30](#_Toc171978480)

[*5.2.1* *Data* 30](#_Toc171978481)

[*5.2.2* *Cleaning* 30](#_Toc171978482)

[*5.2.3* *Case Folding* 31](#_Toc171978483)

[5.2.4 Normalisasi 31](#_Toc171978484)

[*5.2.5* *Tokenizing* 32](#_Toc171978485)

[5.2.6 *Filtering* 33](#_Toc171978486)

[*5.2.7* *Stemming* 34](#_Toc171978487)

[*5.3* Hasil Pembobotan Kata dengan TF-IDF 34](#_Toc171978488)

[5.4 Hasil Pemodelan 35](#_Toc171978489)

[5.5 Hasil Evaluasi 35](#_Toc171978490)

[5.6 Hasil *Integrasi* Model dan Web 38](#_Toc171978491)

[*5.6.1* Hasil *Integrasi* 38](#_Toc171978492)

[5.6.2 Hasil Halaman Branda 39](#_Toc171978493)

[6 BAB VI KESIMPULAN DAN SARAN 40](#_Toc171978494)

[6.1 Kesimpulan 40](#_Toc171978495)

[6.2 Saran 40](#_Toc171978496)

[7 DAFTAR PUSTAKA 41](#_Toc171978497)

[8 LAMPIRAN 43](#_Toc171978498)

# DAFTAR GAMBAR

[Gambar 2. 1 SVM 10](#_Toc171479676)

[Gambar 2. 2 Random Forest 11](#_Toc171479677)

[Gambar 4. 1 Perancangan Proses Diagram Aliran 19](#_Toc171978535)

[Gambar 4. 2 Perancangan Proses Data Selection 20](#_Toc171978536)

[Gambar 4. 3 Perancangan Proses Text Preprocessing 21](#_Toc171978537)

[Gambar 4. 4Perancangan Proses Pembobotan Kata 23](#_Toc171978538)

[Gambar 4. 5 Perancangan Pemodelan 24](#_Toc171978539)

[Gambar 4. 6 Perancangan Pemodelan 25](#_Toc171978540)

[Gambar 4. 7 Perancangan Proses Integrasi Model 26](#_Toc171978541)

[Gambar 5. 1 Hasil Labeling 30](#_Toc171479769)

[Gambar 5. 2 Hasil Data 30](#_Toc171479770)

[Gambar 5. 3Hasil Cleaning 31](#_Toc171479771)

[Gambar 5. 4 Hasil Filtering (Stopword Removal) 33](#_Toc171479772)

[Gambar 5. 5 Hasil Evaluasi SVM 35](#_Toc171479773)

[Gambar 5. 6 Confusion Matrix SVM 37](#_Toc171479774)

[Gambar 5. 7 Halaman Beranda 39](#_Toc171479775)

# DAFTAR TABEL

[Tabel 2. 1 Peneliti Terdahulu 7](#_Toc171479791)

[Tabel 4. 1 Analisis Kebutuhan Fungsional 18](#_Toc171479853)

[Tabel 4. 2 Fitur dan Keterangan 18](#_Toc171479854)

[Tabel 5. 1 Hasil Scraping Data 28](#_Toc171479817)

[Tabel 5. 2 Hasil Case Folding 31](#_Toc171479818)

[Tabel 5. 3 Hasil Normalisasi 32](#_Toc171479819)

[Tabel 5. 4 Hasil Tokenizing 32](#_Toc171479820)

[Tabel 5. 5 Stemming 34](#_Toc171479821)

[Tabel 5. 6 Hasil Akurasi Pemodelan 35](#_Toc171479822)

[Tabel 5. 7 Hasil Evaluasi 38](#_Toc171479823)

# BAB I PENDAHULUAN

## Latar Belakang

Di masa globalisasi yang sedang berlangsung, setiap orang membutuhkan kemampuan untuk menyesuaikan diri dengan inovasi yang ada. Pemanfaatan dan optimalisasi teknologi digital telah melahirkan bentuk baru birokrasi pemerintahan yaitu *Electronic Government* (*EGovernment*) [1]. Oleh karena itu, tidak menutup kemungkinan jika setiap instansi pemerintahan proaktif melaksanakan berbagai program untuk mempercepat transformasi digital. Salah satunya yang dilakukan oleh Direktorat Jenderal Imigrasi, kebijakan pelayanan paspor Indonesia telah mengalami beberapa kali perubahan dan pembaruan sebagai respons terhadap pandemi COVID-19, mulai dari penerapan kebijakan *Eazy Passport* hingga pembaruan kebijakan antrian pengajuan paspor dengan *Mobile Passport* (M-Paspor) [2].

Saat ini aplikasi M-Paspor di Google Play Store telah diunduh oleh pengguna sebanyak lebih dari 1 juta, dan jumlah ulasan 30,8 ribu dengan rating rata-rata ulasan pada nilai 2,3. Rating tinggi yang disertai dengan ulasan positif dan negatif bukan berarti bahwa harapan pengguna aplikasi terpenuhi dan berhenti untuk melakukan evaluasi. Hal itu dikarenakan terdapat beberapa pengguna yang memberikan rating yang tinggi namun ulasan yang diberikan merupakan ulasan negatif. Ulasan pengguna yang tidak sesuai dengan rating yang diberikan diperlukan untuk dilakukan analisis sentimen.

Analisis Sentimen ialah studi komputasi mengenai persepsi, sikap, dan emosi masyarakat terhadap suatu entitas [3]. Entitas tersebut dapat berupa individu, peristiwa, ataupun topik. Oleh karena itu, tujuan dari analisis sentimen yaitu untuk memutuskan apakah suatu entitas mempunyai nilai positif atau negatif. Dalam melakukan analisis sentimen, maka diperlukan pendekatan metode *machine learning*. Dalam hal ini, untuk melakukan klasifikasi peneliti cenderung memakai metode *SVM, Naïve Bayes,* dan *Random Forest* dalam penelitiannya[4]. Pada penelitian dengan kasus yang serupa pula, metode *SVM* dan *Random Forest* memiliki akurasi yang cukup bagus dibandingkan metode lainnya.

Adapun beberapa penelitian yang mengulas perbandingan metode analisis sentimen diantaranya adalah Akhmad Miftahusalam, dkk yang meneliti tentang perbandingan metode *Random Forest* dan *Naive Bayes* dalam melakukan analisis sentimen terhadap ulasan aplikasi BCA Mobile. Penelitian tersebut menghasilkan kesimpulan bahwa klasifikasi teks lebih baik menggunakan metode *Random Forest* dari pada *Naïve* Bayes dengan perbandingan tingkat akurasi 93,93% dan 92,31% [5]. Penelitian lain juga di lakukan oleh Lutfi Budi Ilmawan, dkk yang melakukan perbandingan metode *SVM* dan *Naïve Bayes* pada analisis sentimen ulasan tekstual di *Google Play Store*. Hasil dari penelitian tesebut mendapatkan nilai akurasi sebesar 81,46% untuk metode *SVM,* dan 75,41% untuk nilai akurasi menggunakan metode *Naïve Bayes* [6].

Dengan mempertimbangkan masalah diatas, maka penulis memilih sebuah judul “Perbandingan Metode *SVM* Dan *Random Forest* Terhadap Analisis Sentimen Ulasan Aplikasi M-Paspor”. Tujuan dari penelitian ini adalah untuk melakukan komparasi antara metode *SVM* dan *Random Forest* dalam melakukan analisis sentimen ulasan aplikasi M-Paspor.

## Rumusan Masalah

Dengan mempertimbangkan latar belakang yang ada, maka dibuatlah rumusan masalah sebagai berikut:

1. Bagaimana cara melakukan analisis sentimen terhadap ulasan aplikasi M-Paspor dengan menggunakan metode klasifikasi *SVM* dan *Random Forest*?
2. Metode mana yang terbaik antara *SVM* dan *Random Forest* dalam melakukan analisis sentimen pada ulasan aplikasi M-Paspor?

## Batasan Masalah

Agar penelitian ini dapat berjalan dengan baik, terarah dan terkonsep. Adapun batasan masalah dari penulisan ini adalah sebagai berikut:

1. Penelitian ini menggunakan metode *SVM* dan *Random Forest*.
2. Teknik web scraping digunakan dalam pengumpulan data, dalam hal ini pengumpulan data dilakukan melalui situs web *Google Play Store*.
3. Terdapat 2.787 data yang telah didapatkan dari tahapan scraping pada laman *Google Play Store*.

## Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Dapat mengimplementasikan metode *SVM* dan *Random Forest* pada analisis sentimen terhadap ulasan aplikasi M-Paspor.
2. Untuk mengetahui perbandingan kinerja metode *SVM* dan *Random Forest* dalam melakukan analisis sentimen pada ulasan aplikasi M-Paspor.

## Manfaat Penelitian

Manfaat penelitian ini bagi pengembang aplikasi M-Paspor adalah untuk membantu proses evaluasi dan pemeliharaan sistem yang didapat dari umpan balik ulasan aplikasi yang diterima pada *Google Play Store*, sehingga pengembang dapat meningkatkan kualitas dari aplikasi tersebut.

Sedangkan bagi calon pengguna adalah sebagai bahan pertimbangan sebelum memutuskan untuk memakai aplikasi M-Paspor. Calon pengguna dapat menjadikan analisis sentimen ini sebagai bahan pertimbangan sebelum menggunakan aplikasi M-Paspor.

Dengan menghasilkan data baru, menemukan, atau analisia yang telah dilakukan, diharapkan dapat berkontribusi di bidang ilmu pengetahuan, yang dapat digunakan oleh komunitas akademis dan praktisi di masa depan.

## Metodologi Penelitian

Dalam sebuah penelitian dibutuhkan gambaran dari alur proses yang diperlukan mulai dari tahap awal hingga hasil. Adapun tahapan-tahapan dalam penelitian ini adalah:

1. Identifikasi Masalah

Identifikasi masalah merupakan langkah awal dalam suatu penelitian yang bertujuan untuk mendefinisikan dan menjelaskan masalah yang ingin diselesaikan dengan menggunakan metode penelitian yang tepat.

1. Studi Pustaka

Studi pustaka bertujuan untuk memberikan gambaran umum dan landasan teori terkait topik atau masalah yang diteliti.

1. Pengumpulan Data

Pengumpulan Data, penelitian ini menggunakan data ulasan aplikasi M-Paspor yang didapatkan dari teknik web scraping pada situs web *Google Play Store* sebanyak 2.787 data.

1. Pengolahan Data

Padatahap ini akan dilakukan tahapan data *cleaning*, *labeling* dan *preprocessing* data.

1. Pengujian dan Evaluasi

Hasil dari pengolahan data akan di ujicobakan dengan menggunakan metode *SVM* dan *Random Forest*. Kedua metode tersebut dibandingkan nilai akurasi, presisi, *recall*, dan *f1-score* nya, setelah itu akan dilakukan evaluasi pada hasil tersebut..

## Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

|  |  |
| --- | --- |
| **BAB I** | **Pendahuluan**  Isi dari bab ini adalah latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi dan sistematika penulisan. |
| **BAB II** | **Landasan Teori**  Selain memberikan penjelasan tentang berbagai istilah yang terkait dengan topik penelitian, bab ini memberikan dasar teori yang relevan dengan penelitian yang akan dilakukan. |
| **BAB III** | **Metode Penelitian**  Bab ini berisi tentang penjelasan tahapan-tahapan dalam melakukan penelitian, dalam hal ini yaitu studi literatur, pengumpulan data, pelabelan data, *preprocessing*, pengujian dan evaluasi. |
| **BAB IV** | **Analisis dan Perancangan Sistem**  Di bab ini berisi pembahasan terkait analisa dan perancangan dari penelitian. Pembahasan ditujukan untuk menguraikan kebutuhan-kebutuhan yang diperlukan dalam penelitian. |
| **BAB V** | **Hasil dan Pengujian**  Bab ini menjelaskan tentang hasil analisis perbandingan dari pemodelan yang telah dilakukan, dan melakukan pengujian terhadap model tersebut. |
| **BAB VI** | **Kesimpulan dan Saran**  Selain memberikan penjelasan terkait kesimpulan, pada bab ini juga berisi saran guna memperbaiki, atau mengembangkan penelitian yang dilakukan. |

# BAB II LANDASAN TEORI

## Tinjauan Pustaka

Penelitian ini dilakukan tidak terlepas dari hasil penelitian-penelitian terdahulu yang digunakan sebagai bahan perbandingan dan referensi. Penelitiain ini bereferensi pada penelitian sebelumnya yang memiliki persamaan penelitian yaitu meneliti tentang tingkat sentimen ulasan pengguna aplikasi. Sedangkan perbedaan penelitian adalah dari studi kasus penelitian. Penelitian pertama adalah Sentimen Analisis Review Aplikasi Digital Korlantas Pada *Google Play Store* Menggunakan Metode *SVM* oleh Nanda Ressq Setiawan dan Emil R. Kaburuan, 2023. Penelitian tersebut memiliki kesamaan yaitu menggunakan metode *SVM* dalam melakukan analisis sentimen, namun perbedaanya terletak pada studi kasus yang diteliti yaitu ulasan aplikasi korlantas.

Penelitian kedua adalah Analisis Sentimen Ulasan Aplikasi Dana dengan Metode *Random Forest* oleh Fanka Angelina Larasat, Dian Eka Ratnawati, dan Buce Trias Hanggara, 2022. Penelitian tersebut memiliki kesamaan yaitu menggunakan metode *Random Forest* dalam melakukan analisis sentimen namun perbedaanya terletak pada studi kasus yang diteliti yaitu ulasan aplikasi Dana. Penelitian ketiga adalah Perbandingan Metode *Random Forest* dan Naive Bayes pada Analisis Sentimen Review Aplikasi BCA Mobile oleh Akhmad Miftahusalam, Hasih Pratiwi, dan Isnandar Slamet, 2023. Penelitian tersebut memiliki kesamaan yaitu meneliti tentang perbandingan metode *Random Forest* dalam melakukan analisis sentimen namun perbedaanya terletak pada metode pembanding nya dan studi kasus yang digunakan. Penelitian keempat adalah Perbandingan Algoritma Support Vector Machine dan *Random Forest* untuk Analisis Sentimen Terhadap Kebijakan Pemerintah Indonesia Terkait Kenaikan Harga BBM Tahun 2022 oleh Muhamad Samantri, dan Afiyati, 2023. Penelitian tersebut memiliki kesamaan yaitu meneliti tentang perbandingan metode *SVM* dalam melakukan analisis sentimen namun perbedaanya terletak pada metode pembanding nya dan studi kasus yang digunakan. Penelitian kelima adalah *Sentiment Analysis and Classification of Restaurant Reviews using Machine Learning* oleh Kanwal Zahoor, Narmeen Zakaria Bawany, dan Soomaiya Hamid, 2020. Penelitian tersebut memiliki kesamaan yaitu meneliti tentang perbandingan metode *SVM* dan *Random Forest* dalam melakukan analisis sentimen namun perbedaanya terletak pada studi kasus yang digunakan. Adapun uraian penelitian terdahulu dalam penelitian ini dijabarkan pada tabel berikut:

Tabel 2. 1 Peneliti Terdahulu

|  |  |  |  |
| --- | --- | --- | --- |
| Peneliti | Tahun | Judul | Hasil Penelitian |
| Nanda Ressq Setiawan, dkk. | 2023 | Sentimen Analisis  Review Aplikasi Digital Korlantas  Pada *Google Play Store*  Menggunakan Metode  *SVM*. | Model *SVM* lebih diunggulkan pada skenario rasio data 90:10. Dengan tingkat  nilai akurasi 82%[10]. |
| Fanka Angelina Larasat, dkk. | 2022 | Analisis Sentimen Ulasan Aplikasi Dana dengan Metode *Random Forest*. | Menggunakan metode *Random Forest* dengan rasio data 80:20 didapatkan nilai presisi 84%, *recall* 84%, *F1- Score* 84% dan akurasi  84% [11]. |
| Akhmad Miftahusalam, dkk. | 2023 | Perbandingan Metode *Random Forest* dan Naive Bayes pada Analisis Sentimen Review Aplikasi BCA Mobile. | Setelah dilakukan pengujian dengan skala perbandingan dataset 80:20, disimpulkan metode *Random Forest* menghasilkan prediksi yang lebih baik dengan perbandingan nilai akurasi sebesar 93,93%  untuk *Random Forest,* |

Berdasarkan dari penelitian diatas, maka penelitian ini akan melakukan perbandingan antara metode *SVM* dan *Random Forest* guna mencari metode mana yang lebih baik dan akurat dalam menentukan sentimen ulasan pada aplikasi M-Paspor.

## M-Paspor

Aplikasi M-Paspor adalah aplikasi yang dikembangkan oleh Direktorat Jenderal Imigrasi, yang sebelum nya dikenal dengan nama Aplikasi Pendaftaran Antrian Paspor Online (APAPO). Pada aplikasi tersebut, masyarakat dapat melakukan pendaftaran untuk permohonan paspor baru dan pendaftaran terkait permohonan penggantian paspor[7].

## Analisis Sentimen

Analisis sentimen adalah cabang ilmu dari *natural language program*, yang berfokus pada mengekstrak informasi subjektif dan sentimen dari suatu teks. Dengan sentimen analisis, informasi yang tadinya tidak terstruktur dapat diubah menjadi data yang lebih terstruktur. Tugas utama analisis sentimen adalah mengklasifikasikan teks dalam dokumen, kalimat, atau opini menjadi sisi positif atau negatif [8].

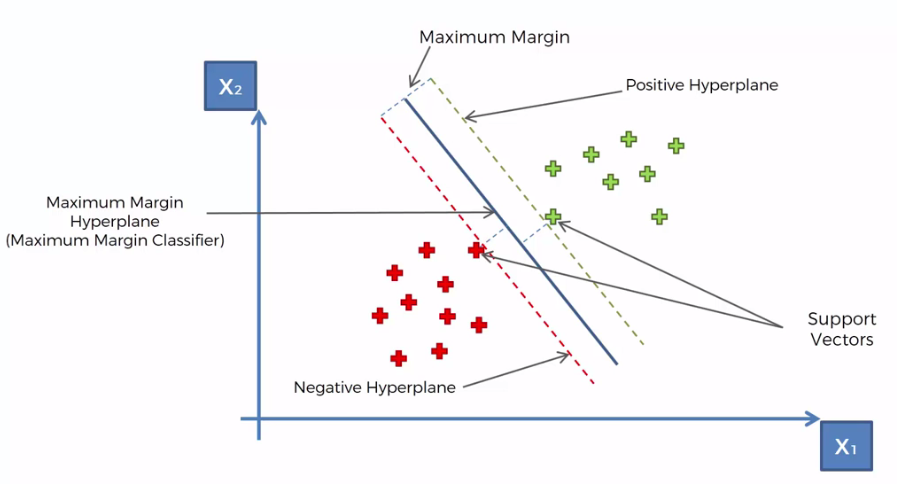
## Ulasan

Menurut Kamus Besar Bahasa Indonesia (KBBI), ulasan merupakan suatu pendapat atau komentar tentang suatu peristiwa. Ulasan adalah kalimat atau text yang memberikan penilaian atau komentar atas karya seseorang [9].

## *Support Vector Machine (SVM)*

Dalam metode *supervised learning*, terdapat metode *Support Vector Machine* (*SVM*) yang biasanya digunakan untuk klasifikasi. Algoritma *SVM* dipilih karena mampu menemukan *hyperplane* terbaik sebagai pemisah. Ini dapat digunakan untuk memisahkan dua kelas data, dalam hal ini kelas positif dan negatif[9]. Berikut adalah konsep dasar dari metode *SVM*:

1. ***Hyperplane***: Dalam konteks *SVM*, *hyperplane* adalah dimensi (n-1) yang digunakan untuk memisahkan data ke dalam dua kelas. Misalnya, dalam kasus dimensi dua, *hyperplane* adalah garis lurus; dalam dimensi tiga, *hyperplane* adalah bidang datar.
2. ***Margin***: *Margin* adalah jarak antara *hyperplane* dan titik-titik terdekat dari kedua kelas. *SVM* berusaha untuk menemukan *hyperplane* yang memiliki *margin* maksimum. Ini berarti *hyperplane* tersebut dapat memaksimalkan jarak antara titik-titik terdekat dari kedua kelas, sehingga meningkatkan kemampuan umum model untuk mengklasifikasikan data baru dengan benar.
3. ***Support Vectors***: *Support vectors* adalah titik-titik data yang berada paling dekat dengan *hyperplane*. Mereka merupakan titik-titik kunci yang mendefinisikan posisi dan orientasi *hyperplane*. Dalam *SVM*, hanya *support* *vectors* yang mempengaruhi penentuan *hyperplane*, sehingga model *SVM* dapat efisien bekerja dengan jumlah fitur atau dimensi yang besar.



Gambar 2. 1 SVM

Rumus perhitungan *hyperplane* dapat ditemukan di bawah ini:

|  |
| --- |
|  |

Dimana:

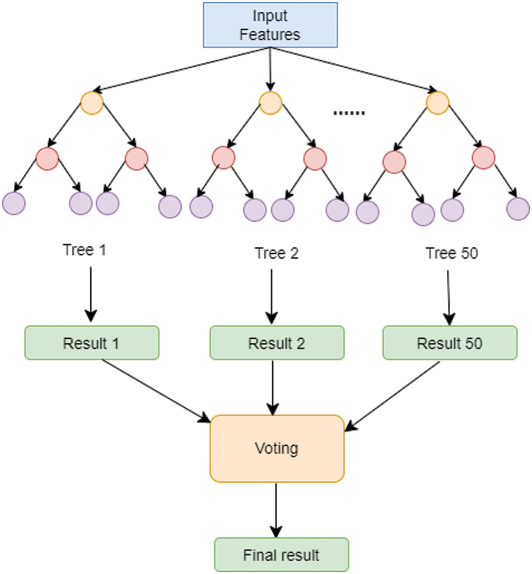
w : Parameter *hyperplane* yang dicari (Garis yang tegak lurus antara garis *hyperplane* dan titik *support vector*).

x : Data input *SVM* (x1 = index kata, x2 = bobot kata).

b : Parameter *hyperplane* yang dicari (Nilai bias).

## *Random Forest*

*Random Forest* merupakan metode machine learning yang berisi gabungan *Decision Tree* untuk dilakukan klasifikasi, dimana untuk membuat keputusan akhir perlu diterapkan *voting mayority* [10]. Metode *Random Forest* dapat bekerja dengan optimal pada ukuran dataset yang sangat besar untuk menyelesaikan tugas klasifikasi, namun memiliki kekurangan yaitu waktu pelatihan yang cukup lama dibanding model lainnya.



Gambar 2. 2 Random Forest

Proses pembentukan setiap pohon keputusan dalam *Random Forest* adalah sebagai berikut:

1. Pemilihan Sampel Acak: Dari dataset pelatihan yang tersedia, sampel data diambil secara acak dengan penggantian (*bootstrap*) untuk membentuk dataset yang berukuran sama dengan dataset pelatihan asli.
2. Pemilihan Fitur Acak: Dari jumlah fitur yang tersedia, subset fitur diambil secara acak untuk digunakan dalam membangun pohon keputusan. Biasanya, jumlah fitur yang diambil setiap kali jauh lebih kecil dari jumlah total fitur yang tersedia.
3. Membangun Pohon Keputusan: Dengan menggunakan sampel data dan subset fitur yang telah dipilih, pohon keputusan dibangun dengan menggunakan algoritma seperti ID3, C4.5, atau CART. Pohon ini dibentuk dengan melakukan pemisahan data berdasarkan fitur-fitur yang paling informatif, dengan tujuan untuk meminimalkan keragaman (*variance*) dalam setiap simpul pohon.
4. Pembentukan Ensambel: Langkah-langkah 1 hingga 3 diulang beberapa kali untuk membentuk kumpulan pohon keputusan. Setiap pohon keputusan dalam *Random Forest* memberikan suara atau memberikan prediksi terhadap kelas atau nilai regresi yang diinginkan.
5. Keputusan Majoritas: Prediksi akhir dalam *Random Forest* diperoleh dengan mengambil mayoritas suara atau rata-rata prediksi dari semua pohon keputusan dalam ensambel.

## *Preprocessing*

Langkah pertama dalam proses pelatihan ialah dengan melakukan preprocessing. Tujuannya adalah untuk memperbaiki data yang tidak konsisten, mengatasi masalah pada data, dan memperbaiki data bermasalah [11]. *Cleaning, case folding, filtering, tokenizing, slangword conversion, stopword removal*, dan *stemming* adalah bagian dari langkah *preprocessing* data.

## Jupyter Notebook

*Jupyter notebook* adalah aplikasi pengembangan interaktif berbasis web, *Jupyter notebook* yang memiliki singkatan *julia* (JU) *python* (PY) dan R sesuai dengan bahasa yang biasa digunakan pada *Jupyter notebook*[12]. *Jupyter notebook* ini bersifat gratis dan yang dapat digunakan untuk visualisasi hasil analisis, dan penentuan keputusan dari pengolahan data.

## Text Mining

Text Mining adalah proses untuk mengekstrak informasi dari data sumber untuk dilakukan analisis, mengelompokan informasi berdasarkan kata-kata, agar dapat mengetahui hubungan dari data sumber lainnya [13].

## *Web Scraping*

Web Scraping merupakan cara untuk mengambil sekumpulan data yang tidak terstruktur dari suatu web secara otomatis dengan perintah program. Hasil dari web scraping biasanya disimpan dalam bentuk spreadsheet atau comma-separated values [14].

## *Confusion Matrix*

Confusion matrix atau matriks kebingungan adalah alat yang digunakan untuk mengukur kinerja model klasifikasi pada data uji yang sudah diketahui hasil sebenarnya. Dalam *confusion matrix* terdapat 4 bagian penting, yaitu:

|  |  |
| --- | --- |
| True Positive(TP) | False Positive(FP) |
| True Negative(TN) | False Negative(FN) |

Dari penggunaan *confusion matrix,* dapat dihitung nilai *precision, accuracy, recall, F1- score*, dengan rumus perhitungan sebagai berikut [15] :

1. Akurasi adalah metrik evaluasi yang mengukur seberapa baik model memprediksi yang benar dari semua pengamatan. Rumus untuk mencari nilai akurasi ialah:

𝐴𝑐𝑐𝑢𝑟𝑎𝑐𝑦 = x100% =

1. Presisi adalah metrik evaluasi yang mengukur prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positf. Rumus untuk mencari nilai presisi ialah:

*Precision* = x100% =

1. *Recall* adalah metrik evaluasi yang mengukur prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Rumus untuk mencari nilai *recall* ialah:

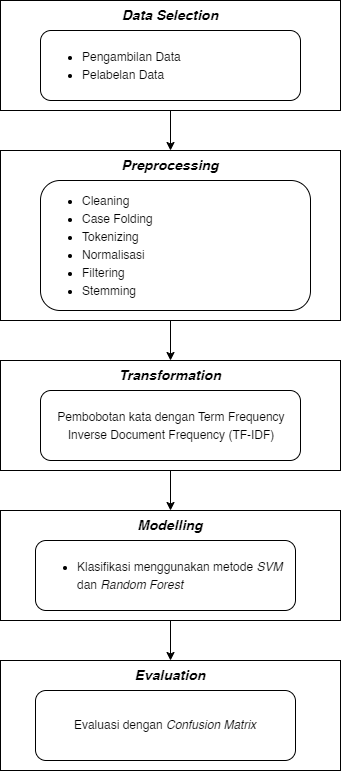
*Recall* = x100% =

1. *F1-Score* merupakan metrik evaluasi yang mencerminkan keseimbangan antara Presisi (*Precision*) dan Sensitifitas (*Recall*). Rumus untuk mencari nilai *f1-score* ialah:

*F1-Score* = =

# BAB III METODE PENELITIAN

Metodologi yang digunakan yaitu *Knowledge Discovery in Database* (KDD) dengan dua pemodelan, pertama *SVM* dan yang kedua *Random Forest*. Dapat dilihat pada gambar dibawahini.



## *Data Selection*

Pada tahap ini berisi pengambilan data dan pelabelan data. Pengumpulan data dalam penelitian ini dengan cara *web* *scaping,* denganmenggunakan library *google-play-scrapper*. Dari proses *web scraping* tersebut, didapatkan data sebanyak 2.787 data. Kemudian data tersebut diklasifikasi ke dalam 2 *class* yaitu Positif dan Negatif berdasarkan nilai rating yang diberikan pengguna. Dimana nilai rating 1-3 dilabelkan sebagai kelas negatif sedangkan 4-5 dilabelkan sebagai kelas positif.

## *Preprocessing*

Preprocessing terdapat beberapa tahapan diantaranya:

* 1. *Cleaning* digunakan untuk proses membersihkan data yang tidak memiliki pengaruh pada proses data mining.
  2. *Case* *Folding* bertujuan untuk merubah semua huruf kapital dalam dokumen menjadi huruf kecil.
  3. *Tokenizing* digunakan untuk memotong kalimat menjadi sebuah kata berdasarkan spasi.
  4. Normalisasi digunakan untuk memperbaiki penulisan kata yang tidak sesuai menjadi sesuai dengan kamus KBBI.
  5. *Filtering* digunakan untuk menghilangkan kata - kata yang tidak bermakna atau kurang penting dengan topik penelitian.
  6. *Stemming* digunakan untuk menghilangkan kata imbuhan dengan menggunakan library sastrawi.

## *Transformation*

Tahapan ini memberikan bobot setiap kata untuk melihat seberapa jauh hubungan kata terhadap dokumen. Bobot dihitung dengan menggunakan *Term Frequency Inverse Document Frequency* (TF-IDF).

## *Modeling*

Pada tahap ini pemodelan menggunakan metode klasifikasi *SVM* dan *Random Forest*, data yang telah melewati tahap *preprocessing* akan dibagi terlebih dahulu menjadi 4 skenario yaitu data *training* dan *testing* dengan perbandingan 90:10, 80:20, 70:30, 60:40*.*

## Evaluasi

Setelah dilakukan proses *modeling* menggunakan metode *SVM* dan *Random Forest*, maka akan dilakukan evaluasi peforma model menggunakan *confusion matrix*. berdasarkan hasil dari pengujian tersebut maka dapat disimpulkan metode mana yang memiliki peforma lebih baik dalam menganalisa sentimen ulasan pengguna aplikas M-Paspor

# BAB IV ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi pembahasan analisis, dan perancangan sistem analisis sentimen. Pembahasan ditujukan untuk menguraikan kebutuhan-kebutuhan dalam pengembangan sistem.

## Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan merupakan proses yang dilakukan sebelum mengerjakan tugas akhir ini tentang Perbandingan Metode *Svm* Dan *Random* Forest Terhadap Analisis Sentimen Ulasan Aplikasi M-Paspor adalah meliputi kebutuhan perangkat keras dan perangkat lunak.

1. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang dibutuhkan dalam mengerjakan penelitian tugas akhir ini agar pengerjaan *text mining* dapat dilakukan karena dalam proses *text mining* 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.

## 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. 1 Analisis Kebutuhan Fungsional

|  |  |  |
| --- | --- | --- |
| **No** | **Kebutuhan** | **Deskripsi** |
| 1. | Aplikasi website. | Aplikasi diharapkan dapat berfungsi dengan semestinya, seperti dapat memasukan input data *text* dan menampilkan hasil pemodelan menggunakan metode *SVM* dan *Random Forest* dari data *text* baru yang di input. |
| 2. | Aplikasi pemodelan | Pemodelan dapat mengklasifikasi sentimen ulasan pada aplikasi M-Paspor menggunakan *SVM* dan *Random Forest*. |

## Analisis Kebutuhan Data

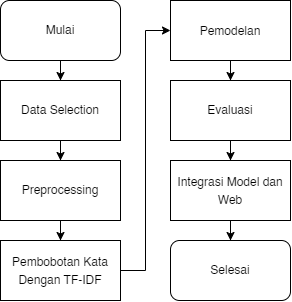
Data yang digunakan merupakan hasil *Scraping* dari *Google Play Store* menggunakan bahasa *python* dengan bantuan pustaka *google play scraper*. Didalam data ini terdapat 2.787 data dan 11 fitur. Atribut atau fitur yang digunakan dalam penelitian ini ada 2, antara lain seperti ditunjukan pada Tabel 4.2.

Tabel 4. 2 Fitur dan Keterangan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Fitur** | **Keterangan** | **Jenis Data** |
| 1. | content | Berisi ulasan pengguna | Kategorikal |
| 2. | score | Berisi *rating* penguna dari 1-5 | Kategorikal |

## Perancangan Proses Diagram Aliran

Pada perancangan ini dijelaskan alur penelitian dan langkah-langkah mengolah data agar hasilnya dapat berfungsi dengan baik dalam manganalisis sentimen.

****

Gambar 4. 1 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 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.*

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

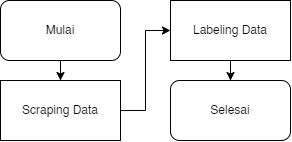
Setelah dilakukan proses pemodelan menggunakan metode *SVM* dan *Random Forest*, maka akan dilakukan evaluasi peforma model menggunakan *confusion matrix*.

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. 2 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*

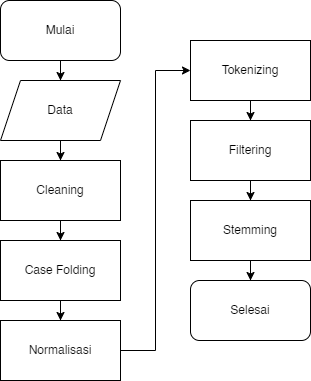
Pengumpulan data dalam penelitian ini dengan cara *web* *scraping,* denganmenggunakan bahasa pemograman *python* serta bantuan pustaka *google-play-scrapper*. Data yang dihasilkan akan disimpan dengan format csv. Penelitian ini memanfaatkan fitur *country=*'id' dari library *google-play-scrapper* supaya 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. 3 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*

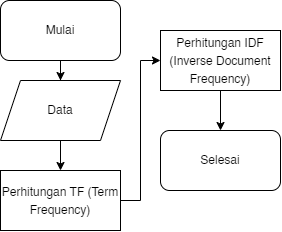
Pada tahapan ini, kata kata yang umum tapi tidak penting akan dihilangkan. Contoh kata-kata tersebut adalah kata depan, kata sambung, kata keterangan dan kata pengganti seperti: “yang”, “ke”, “di”, “sebuah”, “pada”, “oleh”, “ini”, “dari”, dan lain-lain.

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. 4Perancangan 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)*

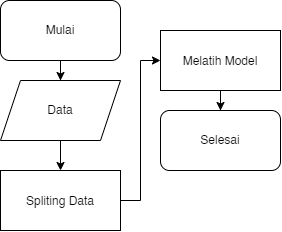
Untuk setiap kata di dalam fitur *content*, hitung berapa kali kata tersebut muncul..

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. 5 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*

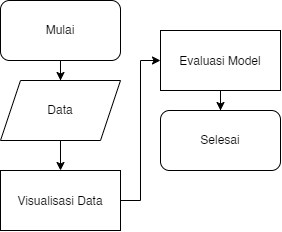
Pada tahapan ini, membagi data *training* dan *testing* denganperbandingan 80:20*.* Proses ini dilakukan dengan bantuan pustaka sklearn.

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. 6 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*

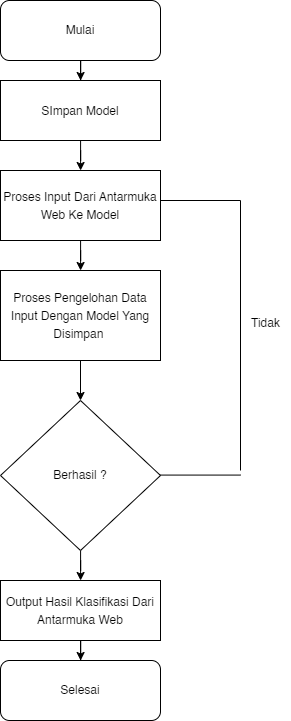
Pada tahapan ini, data yang telah melalui tahapan analisis sentimen akan dipresentasikan ulang dengan harapan mendapatkan informasi baru.

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. 7 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. 1 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 :

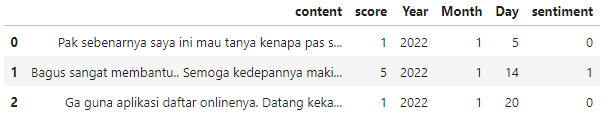
sentimen.append(0)

else:

sentimen.append(1)

df['sentiment'] = sentimen

df.head()



Gambar 5. 1 Hasil Labeling

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

## Hasil *Preprocessing*

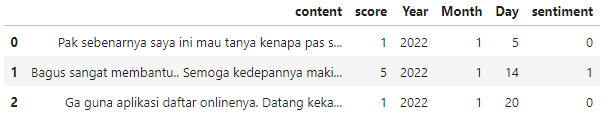
Pada tahap ini akan dilakukan tahapan *preprocessing, preprocessing* adalah proses untuk mempersiapkan data sebelum sebelum dilatih oleh model.

### *Data*

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

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

df = data.copy() #backup df



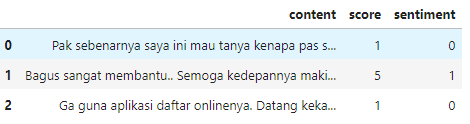
Gambar 5. 2 Hasil Data

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. 3Hasil 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. 2 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. 3 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. 4 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")

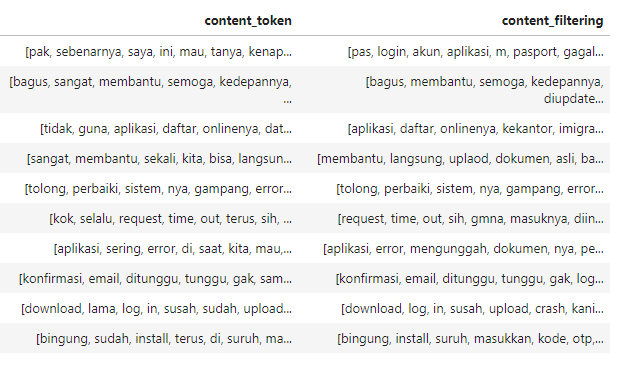
# Extend the list with your own custom stopwords

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. 4 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. 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

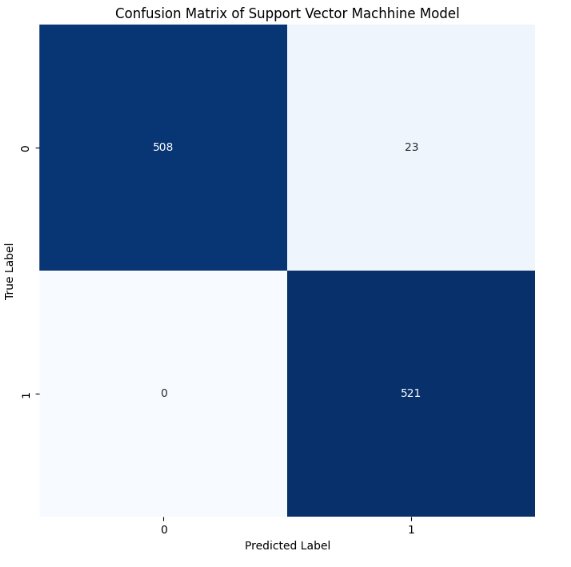
langkah selanjutnya yaitu pemodelan metode *SVM* dan *Random Forest*. Pemodelan dilakukan dengan skenario dataset 80% untuk data latih dan 20% untuk data uji. Berikut adalah hasil pemodelan *SVM* dan *Random Forest* dengan menggunakan skenario data yang telah ditentukan:

Tabel 5. 6 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. 5 Hasil Evaluasi SVM

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

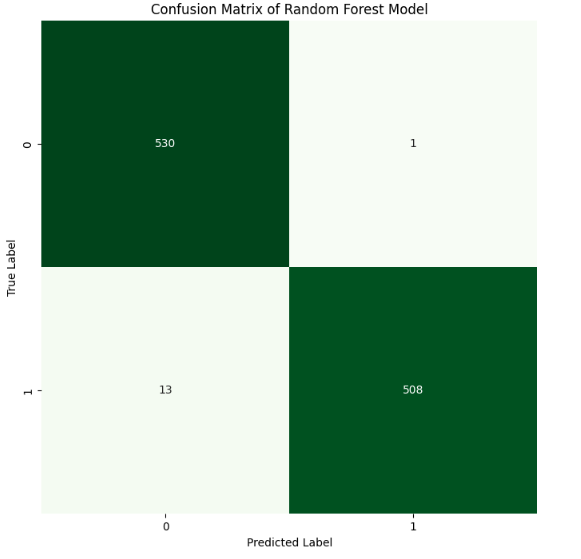
1. *True Positive* (TP), merupakan data *positif* yang diprediksi benar.
2. *True Negative* (TN), merupakan data *negatif* yang diprediksi benar.
3. *False Positive* (FP), merupakan data *negatif* namun diprediksi sebagai data *positif.*
4. *False Negative* (FN), merupakan data *positif* namun diprediksi 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. 6 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. 7 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)

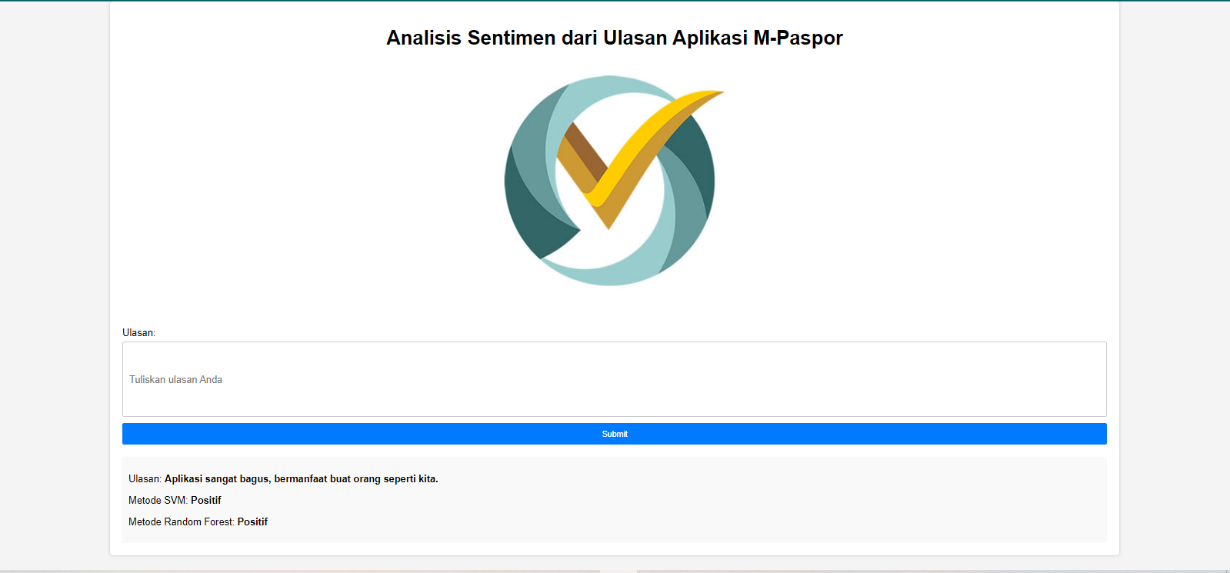
Kode program di atas berfungsi untuk menyimpan pemodelan *SVM* kedalam file *pickle.* Setelah berhasil menyimpan pemodelan dalam file pkl langkah selanjutnya yaitu dengan memanggil file pkl tersebut kedalam aplikasi *website* yang dibangun menggunakan *library* Python yaitu *flask* kedalam file py seperti 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. 7 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

Kesimpulan dari penelitian ini adalah penulis dapat melakukan analisis sentimen terhadap ulasan aplikasi M-Paspor dengan menggunakan metode klasifikasi *SVM* dan *Random Forest*, dan aplikasi yang dibangun dapat memprediksi sentimen dengan baik.

Dari hasil penelitian ini dapat disimpulkan bahwa klasifikasi menggunakan metode *Random Forest* menghasilkan prediksi yang lebih baik dibandingkan dengan metode *SVM*. Hal tersebut ditunjukan dengan nilai *akurasi* yang lebih tinggi.

## Saran

Beberapa saran dari penelitian yang mungkin bisa membantu dalam penelitian selanjutnya:

1. Perluas Dataset: Dalam penelitian ini, dataset yang digunakan memiliki peran penting dalam mempengaruhi kualitas dan keakuratan model. Oleh karena itu, disarankan untuk memperluas dataset dengan mengumpulkan lebih banyak data ulasan pengguna. Hal ini akan meningkatkan representasi data dan memperkuat generalisasi model.
2. Skenario Pengujian Model: Selain skenario dataset pemodelan yang sudah digunakan yaitu 80:20, penelitian selanjutnya dapat menggunakan skenario dataset pengujian yang lainnya. Misalnya, 90:10,70:30, 60:40 dll, agar hasil dapat merepresentasikan pengujian yang lebih baik.

# DAFTAR PUSTAKA

[1] R. P. Wulandari, N. Winarti, E. Darmawan, U. M. Raja, and A. Haji, “Implementasi E-Government Dalam Pemanfaatan Aplikasi Mobile Paspor (M-Paspor) Pada Kantor Imigrasi Kelas Ii Tpi Tanjung Uban,” *Multidisciplinary Indonesian Center Journal (MICJO)*, 2024, doi: 10.62567/micjo.v1i2.49.

[2] Mohamad Wahyudiantoro, C. D. Edwinarta, and R. Syarif, “Fenomena Pembatalan Permohonan Paspor dalam Implementasi Kebijakan M-Paspor pada Kantor Imigrasi Tanjung Perak,” *Journal of Politics and Policy*, vol. 4, no. 2, Dec. 2022, doi: 10.21776/ub.jppol.2022.004.02.02.

[3] W. Medhat, A. Hassan, and H. Korashy, “Sentiment Analysis Algorithms And Applications: A Survey,” *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/j.asej.2014.04.011.

[4] M. Cindo and D. P. Rini, *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS) Literatur Review: Metode Klasifikasi Pada Sentimen Analisis*. 2019. [Online]. Available: https://seminar-id.com/semnas-sainteks2019.html

[5] A. Miftahusalam, H. Pratiwi, and I. Slamet, “Perbandingan Metode Random Forest dan Naive Bayes pada Analisis Sentimen Review Aplikasi BCA Mobile,” 2023.

[6] L. B. Ilmawan and M. A. Mude, “Perbandingan Metode Klasifikasi Support Vector Machine dan Naïve Bayes untuk Analisis Sentimen pada Ulasan Tekstual di Google Play Store,” *ILKOM Jurnal Ilmiah*, vol. 12, no. 2, pp. 154–161, Aug. 2020, doi: 10.33096/ilkom.v12i2.597.154-161.

[7] P. Sari Dia, R. A. Al Karimah, and S. Salsabilla, “Efektivitas Aplikasi M-Paspor Dalam Pelayanan Publik,” *Jurnal Media Administrasi*, no. 9, pp. 169–173, 2024, doi: 10.56444/jma.v9i1.1543.

[8] K. Kevin, M. Enjeli, and A. Wijaya, “Analisis Sentimen Pengunaaan Aplikasi Kinemaster Menggunakan Metode Naive Bayes,” *Jurnal Ilmiah Computer Science*, vol. 2, no. 2, pp. 89–98, Jan. 2024, doi: 10.58602/jics.v2i2.24.

[9] D. Diandra Audiansyah, D. Eka Ratnawati, and B. Trias Hanggara, “Analisis Sentimen Aplikasi MyXL menggunakan Metode Support Vector Machine berdasarkan Ulasan Pengguna di Google Play Store,” 2022. [Online]. Available: http://j-ptiik.ub.ac.id

[10] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, “Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest,” 2022. [Online]. Available: http://j-ptiik.ub.ac.id

[11] F. Septarian and A. Nugroho, “Implementasi Naïve Bayes Classifier Untuk Klasifikasi Sentimen Pengguna Twitter Terhadap Kinerja DPR,” 2023.

[12] J. Wang, L. Li, and A. Zeller, “Better Code, Better Sharing: On the Need of Analyzing Jupyter Notebooks,” in *Proceedings - 2020 ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results, ICSE-NIER 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 53–56. doi: 10.1145/3377816.3381724.

[13] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” *Int J Comput Appl*, vol. 181, no. 1, pp. 25–29, Jul. 2018, doi: 10.5120/ijca2018917395.

[14] U. Analisis, T. Penelitian, and K. Indonesia, “Implementasi Teknik Web Scraping pada Jurnal SINTA.” [Online]. Available: http://sinta2.ristekdikti.go.id/journals/detail

[15] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, “Multi-label Classifier Performance Evaluation with Confusion Matrix,” Academy and Industry Research Collaboration Center (AIRCC), Jun. 2020, pp. 01–14. doi: 10.5121/csit.2020.100801.

BIOGRAFI PENULIS



|  |  |  |
| --- | --- | --- |
| Nama Lengkap | : | Muhammad Afif Taqiyuddin |
| Tempat, Tanggal Lahir | : | Pontianak, 26 Agustus 1999 |
| Jenis Kelamin | : | Laki-laki |
| Agama | : | Islam |
| Alamat | : | Jl. Puskesmas Pal 3, Komp. Batara Alam Indah, Pontianak, Kalimantan Barat, Indonesia |
| Nomor Telpon | : | +62857725665326 |
| Email | : | Muhammad.taqiyuddin@unmuhpnk.ac.id |
| Kata Motivasi | : | “Tidak ada sesuatu yang berharga yang datang dengan mudah” |