

BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Dalam penelitian untuk merancang dan membangun Analisis Sentimen Aplikasi JMO Terkait Aplikasi JMO Terkait Kemudahan Masyarakat dalam Mendapatkan Pelayanan BPJS Ketenagakerjaan Menggunakan Metode *Support Vector Machine* pada platform *google play* memiliki beberapa tahapan dalam perancangannya yaitu:

1. Studi Literatur

Studi literatur adalah langkah awal dalam penelitian ini untuk mendapatkan informasi tentang subjek yang diteliti. Informasi diperoleh dengan membaca dan memahami jurnal ilmiah, karya tulis, dan buku. Teori yang terkait dalam penelitian ini adalah Analisis Sentimen, *Aplikasi JMO*, *Support Vector Machine*, *Confusion Matrix*, serta *TF-IDF*.

2. Pengumpulan Data

Pada tahap pengumpulan data digunakanlah API *google_play_scraper* yang dapat dimanfaatkan untuk memperoleh *raw data* berupa *review* atau ulasan secara utuh dari platform *google play*.

3. Perancangan

Setelah data dikumpulkan, langkah berikutnya adalah perancangan sistem. Tahapan ini dimulai dengan membuat flowchart penelitian, melakukan pra-pemrosesan data, *labelling data* dan membagi data menjadi data training dan data testing.

4. Implementasi Algoritma

Setelah perancangan selesai, langkah selanjutnya adalah melakukan pembobotan dengan TF-IDF, lalu melakukan klasifikasi dengan *Support Vector Machine*.

5. Pengujian

Tahapan ini dilakukan untuk menguji sistem yang telah dibangun dengan melakukan pengujian terhadap code untuk menemukan error atau bug. Pengujian juga dilakukan untuk mengetahui apakah sistem sudah berjalan dengan semestinya. Pengujian ini dilakukan dengan membandingkan kernel pada algoritma SVM dan membagi data menjadi data training dan data testing dengan perbandingan 60:40, 70:30, dan 80:20. Hasil pengujian kemudian dianalisis menggunakan *confusion matrix* untuk menghitung tingkat *accuracy*, *precision*, *recall*, dan *f1-score*. Pengujian ini dilakukan untuk mencari kernel terbaik pada metode SVM dan mengetahui pengaruh perbandingan data training dan data testing dalam analisis sentimen.

6. Dokumentasi

Proses penelitian, perancangan, dan pembuatan aplikasi hingga selesai didokumentasikan secara bertahap, mulai dari pendahuluan hingga kesimpulan dan saran. Dokumentasi ini bermanfaat untuk memberikan informasi dan manfaat bagi penelitian lainnya.

7. Konsultasi

Konsultasi dengan dosen pembimbing bertujuan untuk memperoleh saran dan masukan terkait perbaikan aplikasi dan laporan penelitian.

8. Evaluasi

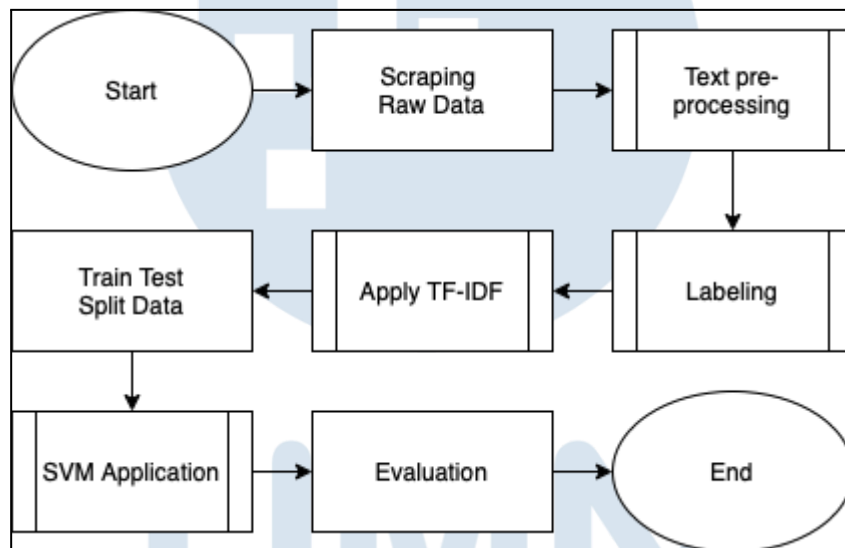
Evaluasi dilakukan untuk mengukur akurasi dan kinerja model SVM.

3.2. Perancangan Sistem

Pada perancangan sistem ini, akan dijelaskan diagram alur sistem yang telah dibuat. Mulai dari diagram alur tersebut meliputi diagram alur gambaran umum perancangan, diagram alur pra-pemrosesan, diagram alur pembobotan TF-IDF, hingga diagram alur klasifikasi dengan *Support Vector Machine*.

3.2.1 Gambaran Perancangan

Langkah-langkah yang diimplementasikan pada perancangan dan penelitian akan disusun dan gambarkan melalui diagram flowchart mulai dari awal hingga akhir.



Gambar 3.1 *Flowchart* Gambaran Perancangan

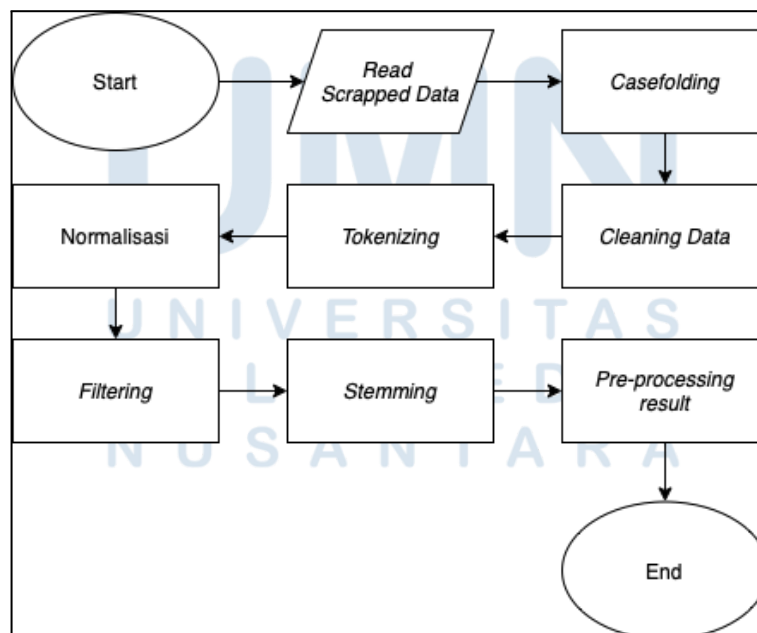
Pada Gambar 3.1 telah terlampir proses perancangan mulai dari crawling data, *text-preprocessing*, *labeling*, *apply TF-IDF*, *train-test split data*, *SVM application* dan hingga *evaluation*.

3.2.2 Scrapping Raw Data

Scrapping Raw Data merupakan suatu metode untuk mengambil informasi dalam sebuah website menggunakan automation program dan menggunakan *Application Programming Interface* (API) sebagai jalur komunikasi dalam mengambil data. Pada penelitian ini crawling data dilakukan menggunakan API *google_play_scraper* untuk mendapatkan tinjauan dari pengguna. Pada penelitian ini *crawling data* dilakukan terhadap aplikasi JMO, data-data yang dikumpulkan hanyalah data tinjauan yang menggunakan bahasa Indonesia mulai dari tahun 2018 hingga 2023.

3.2.3 Text Preprocessing

Tahapan *text-preprocessing* teks adalah proses untuk membersihkan dan menstandarisasi data teks agar menjadi lebih mudah untuk dianalisis. Penelitian menemukan bahwa tahapan pra-pemrosesan teks yang paling efektif adalah dengan melakukan normalisasi terlebih dahulu, kemudian diikuti oleh proses stopword removal dan stemming. Gambar 3.3 menunjukkan tahapan-tahapan *preprocessing* teks, mulai dari *case folding*, *cleaning data*, *tokenizing*, *normalisasi*, *filtering*, hingga *stemming*.



Gambar 3.2 *Text Preprocessing Flowchart*.

1. Case Folding

Case Folding adalah tahapan untuk menyelaraskan teks menjadi huruf kecil, hal ini dikarenakan data yang kita miliki tidak selalu terstruktur dan konsisten dalam penggunaan huruf kapital. Jadi, peran dari case folding adalah untuk menyamaratakan penggunaan huruf.

2. Cleaning Data

Tahapan ini menerapkan seleksi dengan membersihkan teks dari simbol, *emoticon, username, hashtag, link, whitespace, single char, number dan punctuation.*

3. Tokenizing

Tokenizing akan menseparasi kalimat menjadi daftar kata. Hal ini dilakukan untuk tahapan-tahapan berikutnya.

4. Normalisasi

Normalisasi bertujuan untuk memperbaiki kata singkat yang tidak baku menjadi kata dasar yang baku. Contoh penggunaan normalisasi seperti kata "yg" menjadi kata "yang".

5. Filtering

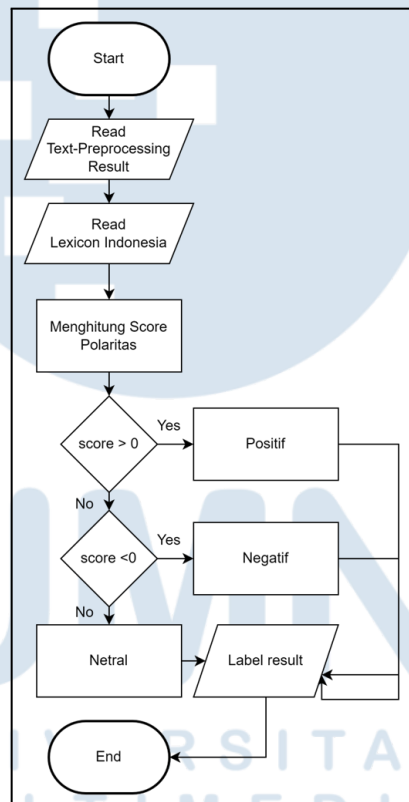
Filtering digunakan untuk mengambil kata-kata yang penting dari hasil *tokenizing*. Kata umum yang biasanya muncul dan tidak memiliki makna disebut dengan *stopword*, kata penghubung seperti dan, yang, serta, setelah, dan lainnya. Penghilangan *stopword* ini dapat mengurangi ukuran *index* dan waktu pemrosesan. Selain itu, juga dapat mengurangi level *noise*.

6. Stemming

Tahapan Stemming berguna untuk menghilangkan kata imbuhan guna mengekstrak bentuk kata dasar. Contoh pengaplikasian stemming adalah kata "memper-mudah" menjadi "mudah".

3.2.4 Labeling Data

Labeling data merupakan pemberian klasifikasi terhadap setiap kalimat yang ada didalam data. Pelabelan ini dilakukan agar pemodelan *machine learning* yang dibuat dapat mengerti dan memahami kalimat teks yang diinput beserta label dari kalimat tersebut. *Labeling data* dalam analisis sentimen dibagi menjadi tiga yaitu positif, negatif dan netral. Pada penelitian ini pelabelan dilakukan menggunakan *automated labeling* dengan kamus *lexicon* berbahasa indonesia beserta bobot dari kata tersebut. Kemudian dihitung nilai polaritas dari setiap kalimat yang ada didalam data.



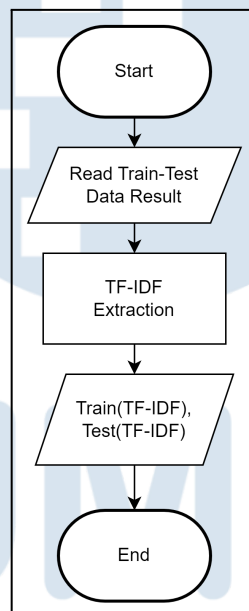
Gambar 3.3 Flowchart Labeling Data.

Gambar 3.3 merupakan flowchart dari labeling data yang dilakukan secara otomatis. Pelabelan dimulai dari membaca data teks yang telah dilakukan *Text-preprocessing* kemudian membaca kata dalam kamus *lexicon* berbahasa Indonesia beserta bobotnya. Selanjutnya membuat fungsi untuk menghitung polaritas dari setiap kalimat yang ada dalam data. Hasil dari polaritas digunakan untuk memberikan label terhadap setiap kalimat jika $score < 0$ akan dimasukkan

ke dalam kelas negatif, jika score > 0 akan dimasukkan ke dalam kelas positif dan jika score = 0 akan dimasukkan ke dalam kelas netral.

3.2.5 Feature Extraction TF-IDF

Sebelum melakukan klasifikasi data menggunakan algoritma *support vector machine*, data terlebih dahulu dibobot menggunakan TF-IDF. Pembobotan ini dilakukan agar algoritma support vector machine dapat memahami kata-kata yang ada dalam data. Pada gambar 3.4, setelah data dibagi menjadi data latih dan data uji, selanjutnya data latih dan data uji akan dibobotkan menggunakan *Feature Extraction TF-IDF*. Hasil pembobotan ini menghasilkan data latih TF-IDF dan data uji TF-IDF.



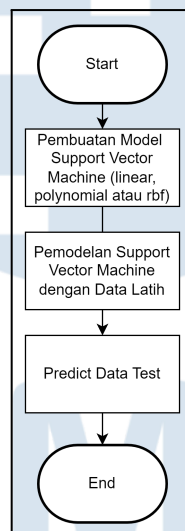
Gambar 3.4 *Flowchart Labeling Data*

3.2.6 Train-Test Split Data

Pembagian data menjadi data latih dan data uji adalah teknik dalam machine learning untuk menilai kinerja model. Data latih memiliki fungsi untuk melatih pemodelan *support vector machine* dan data uji memiliki fungsi sebagai data yang berguna untuk dianalisis hasilnya melalui pemodelan yang dibuat. Hal ini dilakukan untuk mengetahui kinerja serta performa dari pemodelan SVM.

3.2.7 SVM Application

Setelah data teks dibobot menggunakan TF-IDF, langkah selanjutnya adalah menerapkan algoritma *support vector machine* untuk mengklasifikasikan data. Gambar 3.5 adalah *flowchart* dari implementasi algoritma SVM dimulai dari membuat pemodelan SVM dengan kernel *linear, polynomial* atau *rbf*. Setelah model telah berhasil dibuat langkah selanjutnya adalah melakukan pemodelan SVM dengan data latih yang telah dilakukan *feature extraction TF-IDF* hal ini dilakukan untuk memberikan pembelajaran terhadap pemodelan supaya bisa mengerti input kata yang dimasukkan. Dari pemodelan tersebut akan digunakan untuk memprediksi data test yang sebelumnya telah dilakukan *feature extraction*



Gambar 3.5 *Flowchart SVM Application*

3.2.8 Evaluation

Tahap evaluasi memanfaatkan *confusion matrix* untuk mengukur performa klasifikasi data yang telah dilakukan oleh algoritma *support vector machine*. Matriks ini terdiri dari tiga kolom dan tiga baris, yaitu positif, netral, dan negatif. Nilai true berarti label sebenarnya dan prediksi sama, sedangkan nilai false berarti prediksi berbeda dengan label sebenarnya. Hasil matriks kebingungan kemudian digunakan untuk menghitung akurasi, presisi, recall, dan f1-score.