BAB III

METODOLOGI PENELITIAN

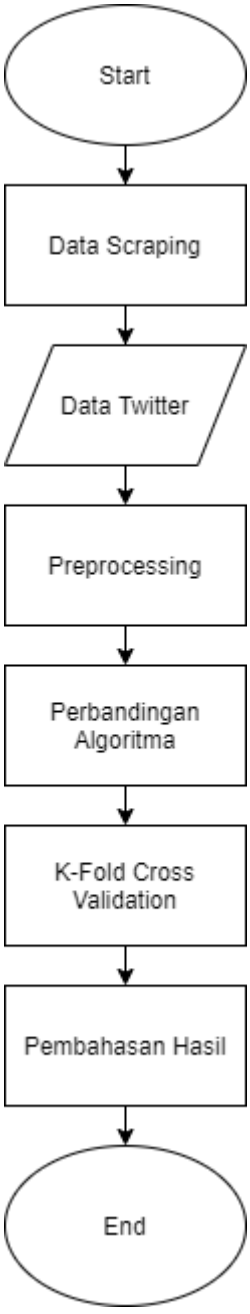
3.1. Objek Penelitian

Penelitian akan berfokus pada analisis sentimen 100 hari kerja Joko Widodo dari tanggal 20 Oktober 2019 sampai dengan 27 Januari 2020. *Tweet* dengan *hashtag*/*trending topic* di *Twitter* yang berkaitan dengan Jokowi. *Tweet* akan digunakan untuk mengetahui sentimen terhadap Joko Widodo positif, negatif, atau netral dan dilakukan perbandingan algoritma *Naïve Bayes*, *KNN*, dan *SVM* untuk mengetahui tingkat akurasinya.

3.2. Teknik Pengumpulan Data

Teknik yang digunakan dalam pengumpulan data adalah dengan melakukan *data scraping* di *Twitter* dengan menggunakan *Python* yang mempunyai fungsi untuk mengambil *historical tweets* dari *Twitter*, penggunaan *Python* sangat membantu untuk pengumpulan data karena untuk pengambilan data menggunakan aplikasi *RapidMiner* dengan *API* dari *Twitter* hanya bisa mengambil *tweets* untuk tujuh hari terakhir saja. Contoh data *tweet* Joko Widodo dan Ma'ruf Amin baru saja dilantik menjadi Presiden dan Wakil Presiden untuk periode 2019-2024 #nasional #Jokowi.

3.3. Alur Penelitian



Gambar 3. 1 *Flowchart* Alur Penelitian

3.3.1. *Data Scraping*

Data scraping dilakukan dengan menggunakan *Python.* Cara *data scraping* yaitu dengan mengisi kriteria data yang dibutuhkan untuk pengambilan data.

Mengunakan *python* dengan *command* yang diperlukan untuk *data scraping*. *Hashtag* yang digunakan untuk *data scraping* adalah #jokowi, #jokowimaruf, #menterijokowi. Ketiga *hashtag* yang terpilih merupakan *trending topic* yang termasuk dalam 100 hari kerja Presiden Joko Widodo dan *hashtag* tersebut juga hampir ada pada *tweet* dengan *hashtag* lain yang berhubungan dengan Joko Widodo. Contoh untuk *data scraping* #kabinetjokowi bisa dilakukan dengan *command twitterscraper* "#kabinetjokowi" --*output* JokoWidodo_11-12Nov.csv --*limit* 1000 --*begindate* 2019-11-10 --*enddate* 2019-11-12 –csv. Dengan *command* tersebut maka *tweet* dengan #kabinetjokowi dari tanggal 10 November 2019 sampai dengan 12 November 2019 akan diambil. Hasil *data scraping* akan tersimpan dalam sebuah file dengan format csv.

3.3.2. Data *Twitter*

Output yang di dapat setelah melakukan *data scraping* disimpan dalam format csv dan mempunyai banyak kolom berisi informasi yang cukup detail seperti tanggal, bahasa, isi *tweet* dan lain-lain. Sebelum data digunakan pada *pre-processing* data perlu disatukan dalam sebuah file baru.

3.3.3. *Pre-processing*

Pre-processing merupakan tahap awal dari *text mining* untuk mengubah data sesuai dengan format yang dibutuhkan. Proses ini dilakukan untuk menggali, mengolah dan mengatur infomasi dan untuk menganalisis hubungan tekstual dari data terstruktur dan data tidak testruktur (Nugroho, 2016).

A. *Cleansing*

Cleansing merupakan tahap untuk menghapus karakter spesial dan tanda baca yang tidak dibutuhkan oleh teks. Tujuannya untuk mengurangi *noise* di *dataset*, contoh karakter yang akan dihilangkan seperti *URL*, *tag*(#), tanda baca seperti (.) koma (,) dan tanda baca lainnya.

B. *Convert Emoticon*

Emosi adalah perasaan intens yang ditujukan kepada seseorang atau sesuatu yang diwakili dengan kombinasi dari huruf, tanda baca, dan angka. Emoji biasanya digunakan orang-orang untuk mengekspresikan perasaan mereka. *Convert emoticon* adalah salah satu cara untuk mengekspresikan perasaan dengan teks.

Contoh untuk kata *convert emoticon* :

Tabel 3. 1 *Convert Emoticon*

Before	After
:) :D :-) :-D B) 8) 8-) ^^ (^_^) =)	senang
:(:-(:< :'(x(;(:x >:x :-x :-# :-&	sedih

C. *Case Folding*

Case folding merupakan tahapan awal pada *pre-processing* yang bertujuan untuk mengubah setiap bentuk kata menjadi sama. Hal ini dilakukan dengan mengubah kata menjadi *lower case* atau huruf kecil.

D. *Tokenization*

Tokenization adalah proses untuk memotong dokumen menjadi pecahan kecil yang dapat berupa bab, sub-bab, paragraf, kalimat, dan kata (token). Hasil dari proses ini disebut token. Dalam beberapa kasus, proses *tokenization* ini juga

dilakukan dengan cara menghilangkan tanda baca yang tidak dibutuhkan. Ada beberapa model *tokenization* yang dapat digunakan seperti *unigram, bigram, trigram* ,dan *n-gram.*

E. *Filtering*

Filtering merupakan tahap untuk menghilangkan kata yang sering muncul tetapi dianggap tidak memiliki arti. *Stopword* merupakan daftar kata umum yang tidak memiliki arti penting. Pada proses ini *stopword* akan dihilangkan dengan tujuan agar *user* bisa lebih fokus terhadap kata lain yang jauh lebih berarti dan penting. Contoh kata *stopword* yaitu di, juga, ini, itu, yang, dan lainnya.

F. *Stemming*

Stemming merupakan tahap untuk mengubah kata imbuhan menjadi kata dasar sesuai dengan aturan bahasa Indonesia yang baik dan benar. Berikut adalah contoh kata dari *stemming* :

Tabel 3. 2 Kata *Stemming*

Before	***After***
Adanya	Ada
awalnya	Awal
membaik	Baik
Sebelum	Belum

G. *Weighting Word*

Weighting word atau pembobotan kata merupakan proses pemberian nilai untuk sebuah kata dalam dokumen teks berdasarkan frekuensi dari kemunculan kata. Salah satu metode paling populer untuk pembobotan kata adalah *TF-IDF*. *TF-*

IDF adalah suatu metode pembobotan kata yang mengabungkan dua konsep yaitu *Term Frequency* dan *Document Frequency.*

3.3.5. Perbandingan Algoritma

1. *Naïve Bayes*

Naïve Bayes adalah *machine learning* yang menggunakan perhitungan probabilitas yang menggunakan konsep pendekatan *Bayesian*. Penggunaan teorema *bayes* dalam algoritma *Naïve Bayes* adalah dengan menggabungkan probabilitas sebelumnya dan probabilitas bersyarat dalam suatu rumus yang dapat digunakan untuk menghitung probabilitas dari masing-masing kalsifikasi yang memungkinan (Tripathi, Lala, & Vishwakarma, 2015).

2. *Support Vector Machine*

Support Vector Machine (*SVM*) merupakan salah satu metode dalam *supervised learning* yang biasanya digunakan untuk klasifikasi (seperti *Support Vector Classification*) dan regresi (*Support Vector Regression*). Dalam pemodelan klasifikasi, *SVM* memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya (Samsudiney, 2019).

3. *K-Nearest Neighbor*

KNN adalah adalah algoritma yang berfungsi untuk melakukan klasifikasi suatu data berdasarkan data pembelajaran (*train data sets*), yang diambil dari k tetangga terdekatnya (*nearest neighbor*). Dengan k merupakan banyaknya tetangga terdekat. *K-nearest neighbor* melakukan klasifikasi dengan proyeksi data pembelajaran pada ruang berdimensi banyak. Ruang ini dibagi menjadi bagian-

bagian yang merepresentasikan kriteria data pembelajaran. Setiap data pembelajaran direpresentasikan menjadi titik-titik c pada ruang dimensi banyak.

3.3.6. *K-Fold Cross Validation*

Setelah data terkumpul, data akan dibagi menjadi *data training* dan *data testing*. Pembagian data akan dilakukan menggunakan metode *k-fold cross validation* untuk menghilangkan kata bias. *K-fold cross validation* membagi dokumen menjadi n bagian.

Dalam satu set eksperimen akan dilakukan sebanyak n percobaan klasifikasi dokumen dengan setiap percobaan menggunakan satu bagian sebagai pengujian data, (n-1) / 2 bagian sebagai label dokumen, dan (n-1) / 2 bagian lain sebagai dokumen yang tidak berlabel akan ditukar setiap percobaan sebanyak n kali. Kumpulan dokumen pertama akan disortir secara acak sebelum dimasukkan ke dalam *fold.* Ini dilakukan untuk menghindari pengelompokkan dokumen dari satu kategori pada *fold*.

3.3.7. Hasil *Label Prediction* dan *Data Visualization*

Setelah proses analisis sentimen selesai, langkah selanjutnya menentukan kualitas dari proses yang telah dilakukan, yaitu mengevaluasi hasil. Proses perhitungan yang telah dilakukan akan diuji dengan parameter *accuracy, precision,* dan *recall.*

Accuracy (A) merupakan angka dari dokumen yang diklasifikasi dengan benar, baik *True Positive* dan *True Negative*. Rumus perhitungan *accuracy* :

$$A = \frac{Total\ Correct\ Classified}{Actual}\ x\ 100\% = \frac{(TP + TN)}{(TP + FP + TN + FN)}\ x\ 100\%$$

Rumus 3. 1 *Accuracy*

Keterangan :

A = Accuracy

TP = True Positive

TN = True Negative

FP = Flase Positive

FN = Flase Negative

Precision (P) adalah seberapa banyak hasil pemrosesan yang relevan terhadap informasi yang dicari. *Precision* bisa juga diartikan sebagai klasifikasi dari *True Positive* dan semua data yang diprediksi sebagai *class positive.* Rumus perhitungan *Precision* :

$$P = \frac{Correctly\ Predicted}{Total\ Predicted}\ x\ 100\% = \frac{TP}{(TP + FP)}\ x\ 100\%$$

Rumus 3. 2 *Precision*

Recall (R) adalah seberapa banyak dokumen yang relevan pada koleksi yang dihasilkan oleh sistem. *Recall* bisa juga diartikan sebagai angka dari dokumen yang memiliki klasifikasi *True Positive* dari semua dokumen termasuk *(False Negative)*. Rumus perhitungan *Recall* :

$$R = \frac{Correctly\ Classified}{Actual}\ x\ 100\% = \frac{TP}{(TP + FN)}\ x\ 100\%$$

Rumus 3. 3 *Recall*

Variabel seperti *TP, TN, FP, dan FN* berasal dari *confusion matrix. TN* adalah *True Negative,* data negatif diklasifikasikan sebagai negatif. *TP* adalah *True*

Positive, data positif diklasifikasikan sebagai positif. *FN* adalah *False Negative*, data positif diklasifikasikan sebagai negatif. *FP* adalah *False Positive*, data negatif diklasifikasikan sebagai positif. Berikut contoh tabel untuk *confusion matrix*:

Tabel 3. 3 *Confusion Matrix*

	Prediction Yes	**Prediction No**
True Yes	*TP*	*FN*
True No	*FP*	*TN*

Setelah perbandingan algoritma selesai akan dilakukan *data visualization*. Dengan adanya *data visualization* akan membantu dalam proses analisis data. *Data visualization* akan menggunakan *line chart* untuk mengetahui pergerakan sentimen. Hasil pergerakan sentimen akan membantu dalam analisis lebih lanjut. Dimana setiap ada pergerkan sentimen yang cenderung naik dan turun dapat diketahui tanggalnya. Adanya informasi setiap tanggal membuat analisis semakin mendalam dan dapat mengetahui hal yang mempengaruhi sentimen pada tanggal tersebut.