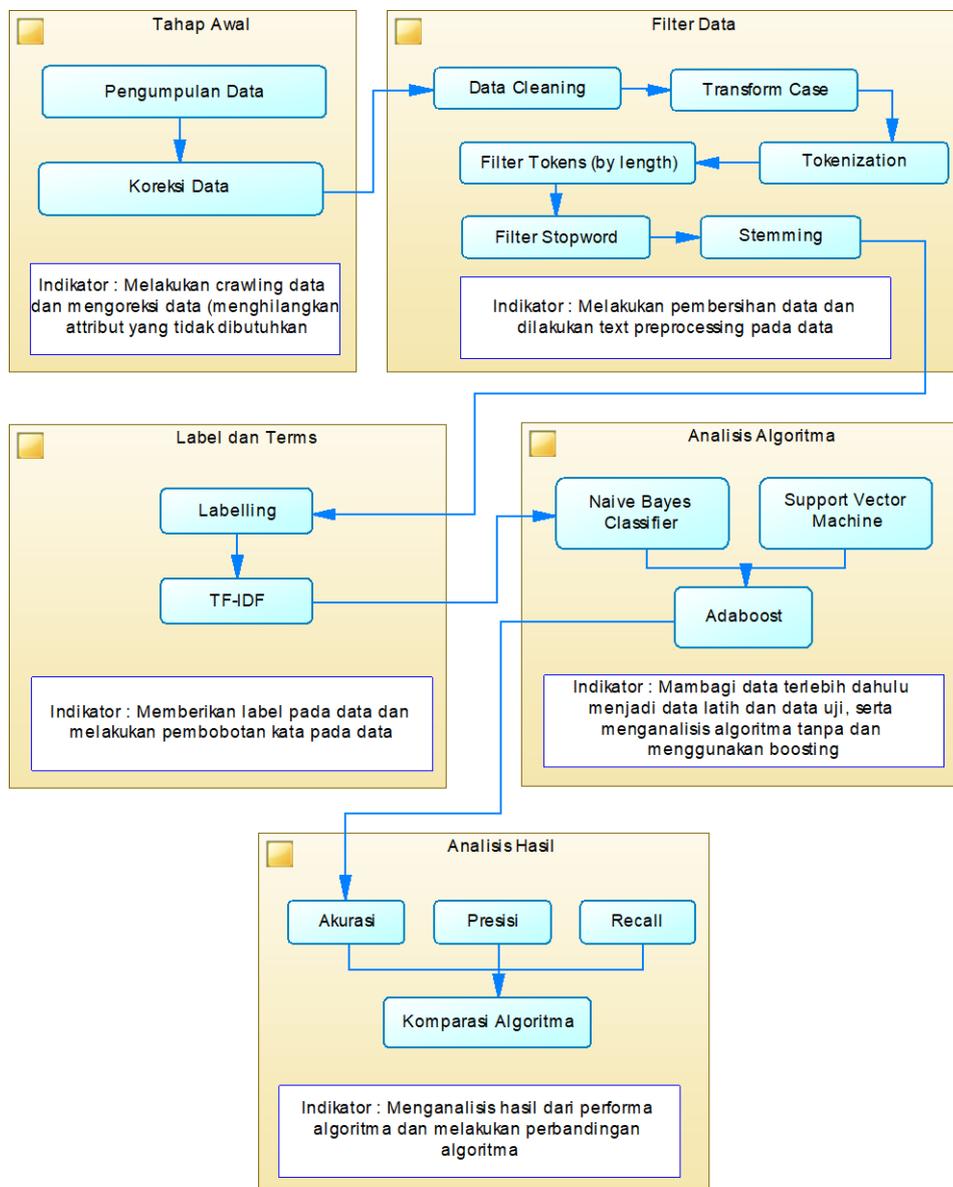


### BAB III

## METODOLOGI PENELITIAN

Metodologi Penelitian adalah sebuah langkah-langkah serta konsep agar bisa memperoleh data yang telah diproses menjadi sebuah informasi yang lebih jelas dan akurat. Berikut merupakan metodologi penelitian yang digunakan pada penelitian ini pada Gambar 3.1 berikut ini.



Gambar 3.1 Metodologi Penelitian

### 3.1 Tahap Awal

Tahap awal ini dilakukan pengumpulan data terlebih dahulu agar mendapatkan dataset. Tahapan awal terdiri dari sebagai berikut :

#### 1. Pengumpulan Data

Penelitian ini tentunya harus ada data yang dipersiapkan terlebih dahulu untuk dapat diolah dan dianalisis. Pengumpulan datanya menggunakan sebuah teknik *crawling* pada media sosial twitter agar bisa mendapatkan data berupa *tweet* orang lain yang berhubungan dengan topik yang dibahas, yaitu berdasarkan kata kunci “kuliah daring” dan “kuliah *online*” menggunakan akses dari pihak twitter dengan memanfaatkan fitur *Twitter API*. *Twitter API* ini dapat memudahkan dalam memperoleh data seperti halnya data pengguna, *tweet*, tanggal *tweet* dan lain-lain.

#### 2. Data Koreksi

Data koreksi dilakukan *import* data ke dalam bentuk CSV terlebih dahulu untuk melakukan koreksi data yang membuang atribut-atribut yang tidak penting dan atribut yang sama agar data bisa langsung diolah pada tahap *preprocessing*.

### 3.2 Filter Data

Tahap selanjutnya dilakukan proses *Filtering* data. Filter Data dilakukan agar performa bisa lebih akurat. Terdapat metode yang dinamakan *Text Preprocessing*. *Text Preprocessing* ini berguna untuk menyeragamkan sebuah teks serta formatnya agar data bisa dipersiapkan dan diolah pada tahap yang selanjutnya. Terdapat beberapa tahapan pada *Text Preprocessing* ini, yaitu diantaranya :

1. *Data Cleaning*

*Data Cleaning* merupakan proses pembersihan data yang sudah diperoleh sebelumnya seperti hanya menghilangkan tanda baca dan lain-lain.

2. *Transform Case*

*Transform Case* merupakan tahapan untuk pemeriksaan pada ukuran setiap karakter dari awal sampai akhir dan merubah semua karakter pada suatu kalimat yang tadinya besar atau kapital menjadi huruf kecil (*lowercase*).

3. *Tokenization*

*Tokenization* ini dilakukan pemecahan string atau pun input terhadap suatu teks dan menghapus kata-kata yang tidak penting seperti halnya URL, *mention*, *hashtag* serta delimiter dan lain-lain.

4. *Filter Tokens (by Length)*

*Filter Tokens (by Length)* dilakukan untuk menghilangkan kata yang mengganggu dan juga menghilangkan kata yang bersifat singkatan dengan cara menghapus kata berdasarkan minimal panjang karakter pada setiap kata atau pun token yang minimal 4 karakter pada setiap katanya.

5. *Filter Stopwords*

*Filter Stopwords* dilakukan penghapusan kata yang tidak memiliki arti dan kurang penting serta kata yang tidak memiliki makna tersendiri berdasarkan data *stopword*.

## 6. *Stemming*

*Stemming* dilakukan pencarian kata dasar untuk merubah kata-kata menjadi kata yang berbentuk dasar dan menghilangkan semua imbuhan pada kata-katanya.

### **3.3 *Label dan Terms***

Tahap selanjutnya ialah proses *Label* dan *Terms* yang merupakan tahap pengambilan label dari data yang telah dipersiapkan sebelumnya menggunakan *text polarity* yang nantinya akan menghasilkan kelas sentimen berupa positif, netral atau pun negatif. *Terms* merupakan proses pembobotan dengan mengubah data tekstual menjadi data numerik menggunakan TF-IDF (*Term Frequency – Inverse Document Frequency*) yang bertujuan untuk menghitung bobot pada suatu kata yang akan digunakan sebagai fitur.

#### 1. *Labelling*

*Labelling* merupakan proses pelabelan terlebih dahulu dengan menggunakan *library* dari *python* yaitu *Textblob*. *Library* ini memudahkan untuk memberikan label secara otomatis dalam jumlah data yang banyak.

#### 2. TF-IDF

TF-IDF merupakan proses perhitungan bobot pada suatu kata yang terdapat pada dokumen dengan mengubah data teks menjadi numerik.

### **3.4 Analisis Algoritma**

Analisis algoritma berguna untuk mengolah datanya. Algoritma yang digunakan adalah *Naïve Bayes Classifier* dan *Support Vector Machine*. Tahap

Analisis Algoritma dilakukan pembagian data menjadi data latih (*training*) 80% dan data uji (*testing*) 20% dilakukan pada dua proses yaitu :

1. *Naïve Bayes Classifier* tanpa Adaboost

Penerapan algoritma *Naïve Bayes Classifier* tanpa menggunakan metode *boosting* yang nantinya akan dilihat kinerja dari algoritma ini berupa akurasi yang didapatkan.

2. *Support Vector Machine* tanpa Adaboost

Penerapan algoritma *Support Vector Machine* tanpa menggunakan metode *boosting* yang nantinya akan dilihat kinerja dari algoritma ini berupa akurasi yang didapatkan dan dilakukan perbandingan.

3. *Naïve Bayes Classifier* dengan Adaboost

Penerapan algoritma *Naïve Bayes Classifier* dengan metode *boosting* yaitu Adaboost dilakukan untuk bisa melihat kinerja dari algoritma yang digunakan serta metode *boosting* yang digunakan berupa akurasi yang nantinya Adaboost ini dapat meningkatkan ketelitian dalam proses klasifikasi dan prediksi dengan cara membangkitkan kombinasi yang dipilih.

4. *Support Vector Machine* dengan Adaboost

Penerapan algoritma SVM dengan metode *boosting* yaitu Adaboost dilakukan juga untuk bisa melihat kinerja atau performa dari algoritma yang digunakan ditambah dengan metode *boosting* yang akan meningkatkan akurasi dan nantinya akan dilakukan perbandingan.

### 3.5 Analisis Hasil

Analisis Hasil akan menampilkan hasil dari performa algoritma yang digunakan, yaitu berupa akurasi yang didapatkan, presisi, *recall* dan akan dilakukan juga evaluasi model serta prediksi data yang telah diklasifikasi menggunakan algoritma *Naïve Bayes Classifier* tanpa dengan *boosting* Adaboost, *Support Vector Machine* tanpa dengan *boosting* Adaboost, *Naïve Bayes Classifier* dengan *boosting* Adaboost dan *Support Vector Machine* dengan *boosting* Adaboost yang nantinya akan dilakukan perbandingan atau komparasi algoritma. Algoritma mana yang memiliki nilai akurasi tertinggi antara algoritma *Naïve Bayes Classifier* dan *Support Vector Machine* tanpa *boosting* serta menggunakan *boosting* Adaboost.