BAB 2
LANDASAN TEORI

2.1 Kesalahan Tik

Kesalahan tik biasanya terjadi pada saat memakai keyboard atau *smartphone*[14], kesalahan ini dapat mengakibatkan makna asli dari teks yang ditulis menjadi sulit untuk dimengerti oleh pembaca teks tersebut. Terdapat beberapa kesalahan tik yang akan diteliti lebih lanjut pada penelitian ini yaitu:

1. Kesalahan penggantian huruf, artinya mengganti atau menambahan huruf dalam satu kata yang memiliki kemiripan ketika dilafalkan misalnya kata ”saya” ditulis menjadi ”sayaa”.
2. Kesalahan pemformatan huruf atau keterbalikan kata, misalnya saja pada kata ”kursi” ditulis menjadi ”krusi” sehingga memiliki makna yang berbeda dari makna asilnya.
3. Kesalahan penghilangan huruf, misalnya ”sepuluh” ditulis ”spuluh”. Terjadi penghilangan huruf ”e” dalam kata ”spuluh”.

2.2 Media Berita Elektronik

Media berita elektronik merupakan salah satu sumber informasi yang paling banyak diakses. Hampir setiap saat ada berita baru yang dimuat ke dalam artikel media berita elektronik dan juga terdapat beberapa artikel yang terkadang dimuat ulang karena terdapat kesalahan dapat menyebabkan misinformasi kepada pembacanya [15]. Media Berita Elektronik biasanya dapat berupa halaman web ataupun aplikasi seluler yang dapat diakses melalui jaringan internet [16], media berita elektronik juga selalu memperbarui atau menambahkan artikel terbaru secara terkini.

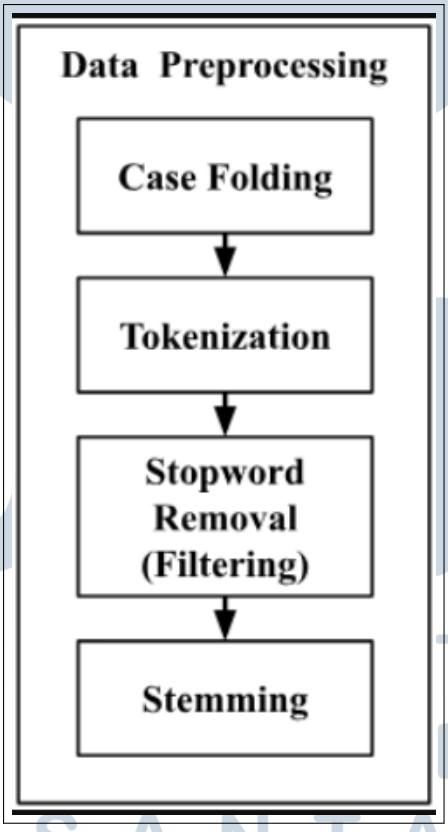
2.3 Natural Language Processing

Natural Language Processing merupakan salah satu bagian dari pembelajaran mesin yang dapat mengelola teks layaknya manusia, *Natural Language Processing* mengubah teks yang ada dalam suatu kalimat kedalam

bentuk pohon yang nantinya dapat digunakan untuk melakukan klasifikasi pada teks tersebut [17]. Terdapat beberapa aplikasi berbasis (NLP) *Natural Language Processing* yang telah dibuat dengan banyaknya data teks yang ada di internet seperti *Chatbot* yang mampu melakukan percakapan melalui teks, mesin pencarian yang dapat menampilkan informasi yang diminta berdasarkan masukkan teks dari pengguna dan juga klasterisasi dokumen untuk mengelompokan dokumen berdasarkan klaster.

2.4 Text Preprocessing

Text Preprocessing dilakukan untuk mengolah data teks yang akan digunakan ke dalam bentuk yang dapat diterima atau diolah oleh algoritma yang akan digunakan. Terdapat beberapa tahapan yang dapat dilakukan pada *Text Preprocessing* sebagai berikut:



Gambar 2.1. *Text Preprocessing* [1]

1. Case Folding, yakni mengubah semua huruf menjadi huruf kecil sehingga dapat melakukan pengecekan tanpa perlu khawatir adanya kata yang sama namun memiliki huruf besar dan huruf kecil [18].

2. Tokenizing, memisahkan kata pada kalimat menjadi kata yang terpisah.

3. Stopword Removal, menghapus kata yang tidak memiliki makna.

4. Stemming, mengubah kata yang memiliki imbuhan ke dalam bentuk dasar kata tersebut [19].

2.5 Multinomial Naive Bayes

Multinomial Naive Bayes merupakan salah satu algoritma yang digunakan dalam klasifikasi teks.Algoritma ini menggunakan teorema Bayes untuk menghitung probabilitas dari suatu kata yang terdapat pada suatu kelas dan pada *Multinomial Naive Bayes* sebuah fitur teks berdiri secara independen satu sama lain [20].

Dalam implementasinya *Multinomial Naive Bayes* membutuhkan data dalam jumlah besar. Walaupun begitu algoritma *Multinomial Naive Bayes* dapat melakukan klasifikasi fitur dengan data yang besar dalam waktu yang cukup singkat dan bisa dibandingkan performanya dengan algoritma pemrosesan teks lainnya yang memiliki algoritma yang lebih kompleks [21].

Tahapan *Multinomial Naive Bayes* dapat dijabarkan sebagasi berikut:

1. Menghitung probabilitas kelas pada artikel atau dokumen yang akan digunakan

$$P(c|d) = P(c)\prod_{k=1}^{n} P(t_k|c) \tag{2.1}$$

   - $P(c|d)$ adalah probabilitas artikel d berada dikelas c.
   - $P(c)$ adalah probabilitas kelas c dari keseluruhan artikel. $P(C)$ didapatkan dengan menggunakan rumus berikut:

$$P(c) = \frac{N_c}{N}$$

     - N_c adalah jumlah artikel pada kelas c
     - N adalah jumlah artike secara keseluruhan
   - tk adalah kata ke-k pada artikel d
   - $P(tk|c)$ adalah probabilitas kata ke-k pada artikel dengan kelas c. $P(tk|c)$ didapatkan dengan menggunakan rumus berikut:

$$P(tk|c) = \frac{n_{tk,c} + 1}{\sum_t (n_{t,c}) + |V|}$$

- $n_{tk,c}$ adalah jumlah kemunculan kata tk dalam artikel yang termasuk dalam kelas c
- $\sum_t (n_{t,c})$ adalah jumlah kata dalam semua artikel yang termasuk dalam kelas c
- $|V|$ adalah jumlah kata unik

Pada perhitungang *maximum likelihood* pada rumus $P(tk|c)$ dilakukan penambahan nilai 1 atau *Laplace smoothing* untuk menghindari probabilitas 0 pada data *training* dan juga untuk mengatasi *overfitting*.

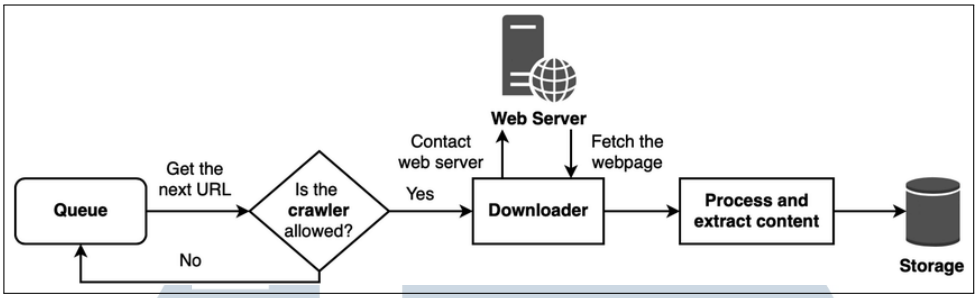
2. Menentukan kelas terbaik dengan mencari *maximum a posteriori* (MAP)

$$c_{\text{map}} = \arg\max_{c \in \mathscr{C}} \hat{P}(c) \prod \hat{P}(t_k|c) \tag{2.2}$$

- c_{map} adalah kelas yang memiliki probabilitas tertinggi
- *argmax* adalah argumen yang mencari kelas c untuk mendapatkan hasil maksimal dari $P(c|d)$
- C adalah himpunan beberapa kelas c
- $P(c)$ adalah probabilitas kelas c dari keseluruhan artikel
- $P(tk|c)$ adalah probabilitas kata ke-k pada artikel dengan kelas c.

2.6 Web Crawling

Web Crawling, yakni proses pengambilan data melalui *website* secara manual dengan tujuan mengambil data secara terstruktur sehingga nantinya dapat dikelola [22]. Dalam melakukan *Web Crawling* perlu diperhatikan apakah data ”kata” yang didapat sudah sesuai dengan standar PUEBI(Pedoman Umum Ejaan Bahasa Indonesia) dan perlu dilakukan pengecekan secara manual maupun di otomatisasi menggunakan kode pemrograman.

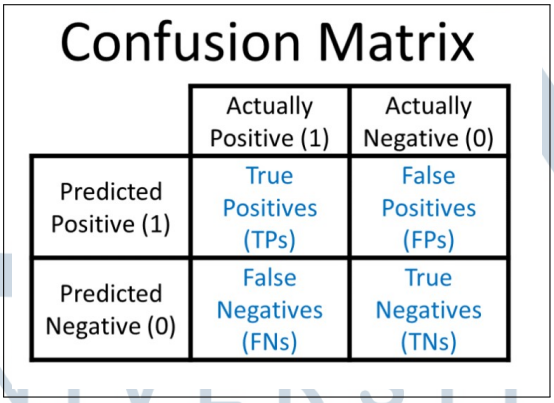


Gambar 2.2. Web Crawling [2]

2.7 Confusion Matrix

Confusion Matrix adalah table yang berisi 4 buah variabel yang digunakan untuk mengukur peforma efektifitas suatu model [23]. Terdapat 4 buah variabel atau kelas yang ada [24], yakni :

1. TP adalah *true positives*, jumlah kelas aktual yang berlabel positif.
2. TN adalah *true negatives*, jumlah kelas negatif yang berlabel benar.
3. FP adalah *false positives*, jumlah kelas negatif yang salah dilabeli oleh model.
4. FN adalah *false negatives*, jumlah kelas positif yang salah dilabeli oleh model.



Gambar 2.3. Confusion Matrix
source: [25]

- Akurasi, Akurasi adalah jumlah presentase kelas yang diklasifikasikan dengan benar oleh model. Untuk menghitung akurasi dapat menggunakan rumus:

$$\frac{TP+TN}{TP+TN+FP+FN} \tag{2.3}$$

- Presisi, Presisi adalah jumlah label positif yang nilai aslinya adalah benar.

$$\frac{TP}{TP+FP} \tag{2.4}$$

- *Recall*, *Recall* adalah jumlah presentase kelas positif yang dilabeli sebagai positif

$$\frac{TP}{TP+FN} \tag{2.5}$$

- *F1-scores*, *F1-scores* adalah nilai rata-rata dari presisi dan *Recall*

$$\frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN} \tag{2.6}$$