BAB 2
TINJAUAN PUSTAKA

2.1 Peluluhan Kata

Peluluhan kata adalah salah satu subset dari morfofonemik, yang merupakan pertemuan antara morfem yang satu dengan morfem yang lainnya (kata atau suku kata) dan menyebabkan perubahan huruf atau fonem [8]. Kaidah peluluhan kata berawalan "k", "p", "s", dan "t" terbagi berdasarkan konsonan kata dasarnya, yaitu konsonan tunggal dan ganda. Jika kata dasarnya berawalan konsonan tunggal, maka kata dengan imbuhan *me-* dan *pe-* akan luluh. Sedangkan kaidah kebahasaan yang benar untuk peluluhan kata dengan gugus konsonan (huruf konsonan ganda) adalah kata dasar dengan imbuhan *me-* dan *pe-* tidak akan luluh [9].

Adapun imbuhan *me-* dan *pe-* pada kata tersebut akan berubah sesuai kata dasar dengan awalan fonem tertentu, yaitu pada awalan "k" akan menjadi *meng-*, awalan "p" menjadi *mem-*, awalan "s" menjadi *meny-*, dan awalan "t" menjadi *men-* . Terdapat pengecualian pada kata dasar dengan gugus konsonan yang berawalan huruf "s", yaitu jika imbuhan *meny-* akan menjadi *men-* untuk kemudahan pelafalan. Sebagai contoh, kata dasar "sponsor" yang memiliki gugus konsonan "sp" dan berawalan huruf "s", maka bentuk benarnya adalah "mensponsori", bukan "meny-sponsori". Contoh lainnya yaitu kata dasar "produksi" yang memiliki gugus konsonan "pr" menjadi "memproduksi", kata dasar "kritik" dengan gugus konsonan "kr" menjadi "mengkritik", dan kata dasar "traktir" dengan gugus konsonan "tr" menjadi "mentraktir" [8, 15].

2.2 *Natural Language Processing*

Natural Language Processing (NLP) merupakan salah satu area bidang ilmu komputer yang meneliti bagaimana membangun interaksi antara komputer dengan manusia yang mudah dimengerti dan efisien, dengan mesin yang menerima sintaks berupa bahasa manusia dan memprosesnya untuk kemudian memberikan respon ke pengguna untuk melakukan berbagai pekerjaan. NLP menjadi semakin penting karena dapat membantu membangun model yang mengambil potongan informasi berupa suara, teks, atau keduanya sebagai *input* dan memanipulasinya sesuai dengan algoritma di dalam komputer [3].

Dasar dari NLP terletak pada beberapa disiplin ilmu, mulai dari ilmu

komputer dan informasi, linguistik, matematika, kecerdasan buatan dan robotika, psikologi, dan sebagainya. NLP dapat diimplementasikan dalam berbagai bidang, seperti penerjemah mesin, pemrosesan dan peringkasan teks bahasa alami, antarmuka pengguna, pengenalan suara, dan sebagainya [16]. .

2.3 *Text Preprocessing*

Text preprocessing merupakan tahap yang penting dilakukan sebelum membangun model NLP, yaitu mengolah data teks untuk mengurangi ukuran kosakata dengan menghapus bagian tidak penting pada data atau *noise*. Dengan melakukan *text preprocessing*, maka ukuran data teks menjadi berkurang dan algoritma pembelajaran mesin yang akan digunakan menjadi lebih efektif dan efisien karena data sudah bersih [17].

Terdapat beberapa langkah utama dalam *text preprocessing* yang dijelaskan pada poin-poin berikut [18].

1. *Case Folding*: merupakan tahap mengubah huruf kapital pada teks menjadi huruf kecil.

2. *Tokenizing*: merupakan tahap pemecahan teks menjadi kalimat atau kata tergantung kebutuhan analisis, yang kemudian disebut "*token*".

3. *Stop-Word Removal*: merupakan tahap untuk menghilangkan kata-kata yang tidak penting atau tidak memiliki banyak makna, seperti "dan", "adalah", dan sebagainya yang tidak banyak berkontribusi terhadap pemahaman konten teks secara keseluruhan. Tahap ini membantu mengurangi dimensi teks sehingga meningkatkan efisiensi model.

4. *Stemming*: merupakan tahap *mapping* dan pemecahan bentuk kata menjadi bentuk dasarnya, seperti menghilangkan imbuhan sehingga teks menjadi lebih sederhana dan mengurangi variasi kata dengan makna dasar yang sama.

2.4 Damerau-Levenshtein Distance

Damerau-Levenshtein Distance adalah algoritma yang dapat digunakan untuk mencari jarak antara *string* target dengan *string* sumber sehingga menghasilkan jarak *edit* antara dua *string* tersebut [19]. *Damerau-Levenshtein Distance* juga didefinisikan sebagai jumlah minimum dari *substitution*, *insertion*,

deletion, dan *transposition* dari karakter yang berdekatan yang dibutuhkan untuk mentransform sebuah *string* menjadi yang lain. Algoritma ini dapat diimplementasikan untuk mendeteksi kesalahan penulisan, *data mining*, *clustering*, deteksi virus pada perangkat lunak, dan sebagainya [20]. Algoritma ini dapat didefinisikan melalui rumus berikut [21].

$$f_{a,b}(i,j) = \min \begin{cases} 0 \quad if i = j = 0 \\ f_{a,b}(i-1,j)+1 \quad \text{if } i > 0 \\ f_{a,b}(i,j-1)+1 \quad \text{if } j > 0 \\ f_{a,b}(i-1,j-1)+1 \left(a_i \text{ and } b_j \text{ are not equal }\right) \quad \text{if } i,j > 0 \\ f_{a,b}(i-2,j-2)+1 \quad \text{if } i,j > 1 \text{ and } a_{i-1} = b_j \text{ and } a_i = b_{j-1} \end{cases} \tag{2.1}$$

Jarak minimum antara dua *string* yaitu a dan b dapat didefinisikan menggunakan Rumus 2.1, dengan i dan j merepresentasikan panjang *prefix* dari *string* a dan b berdasarkan operasi penyuntingan seperti *deletion*, *insertion*, *substitution*, dan *transposition*. Rumus 2.1 tersebut menunjukkan jarak minimum yang diperlukan untuk mengubah awalan (*prefix*) dari panjang i pada *string* a menjadi awalan dari panjang j pada *string* b dan dilakukan secara rekursif.

Ketika kedua string kosong (i dan j bernilai nol), maka jaraknya adalah nol. Pada tahap *deletion*, jika $i > 0$ maka jarak ditentukan dengan menghapus karakter terakhir dari awalan, dan pada tahap *insertion*, jika $j > 0$ maka jarak ditentukan dengan menambahkan karakter ke string b. Pada operasi *substitution*, jika $i, j > 0$ lebih besar dari nol dan karakter terakhir dari kedua *prefix* berbeda, maka jarak ditentukan dengan mengganti karakter di a dengan karakter di b atau sebaliknya sehingga jarak bertambah satu. Sedangkan jika a_i dan b_j sama, maka jarak yang dihasilkan sama dengan jarak dari awalan sebelumnya. Pada operasi *transposition*, jika $i, j > 1$ dan karakter a_{i-1} sama dengan b_j serta a_i sama dengan b_{j-1}, maka jarak minimum diperoleh dengan menukar kedua karakter tersebut.

2.5 *Confusion Matrix*

Confusion Matrix merupakan salah satu metrik yang dapat digunakan untuk mengukur performa model *machine learning* yang dapat menentukan langsung baik atau buruknya performa model berdasarkan hasil distribusi nilai pada sel di *confusion matrix* [22]. Hasil pengukuran ini dapat direpresentasikan menggunakan 4 istilah yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False*

Negative (FN), yang dapat direpresentasikan dalam tabel seperti Tabel 2.1 berikut [23].

Tabel 2.1. Tabel *Confusion Matrix*

	Actual Positive	Actual Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Kemudian, dari hasil *confusion matrix* tersebut dapat dilakukan berbagai evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score* dengan rumus sebagai berikut [24].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.2}$$

$$Precision = \frac{TP}{TP+FP} \tag{2.3}$$

$$Recall = \frac{TP}{TP+FN} \tag{2.4}$$

$$F1score = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.5}$$