BAB 2
LANDASAN TEORI

2.1 Bahasa Indonesia

Bahasa adalah cermin identitas suatu bangsa yang digunakan saat berinteraksi, menjadi alat utama dalam komunikasi manusia. Bahasa Indonesia, yang digunakan di seluruh Indonesia, memainkan peran penting dalam berbagai aspek kehidupan sebagai bahasa persatuan. Untuk memastikan kejelasan dan kemudahan pemahaman, penting bagi Bahasa Indonesia untuk memiliki struktur yang jelas dan mengikuti aturan yang didefinisikan oleh kata baku [22].

Bahasa Indonesia yang dianggap baku digunakan oleh individu yang terdidik dan dijadikan standar untuk penggunaan bahasa yang benar. Standar bahasa ini ditandai oleh karakteristik dinamis dan intelektual. Dinamisme bahasa mengacu pada kemampuannya untuk mengikuti aturan yang tetap namun juga menerima perubahan yang sistematis. Bahasa standar juga memiliki kemampuan untuk mengekspresikan pemikiran yang kompleks dalam berbagai bidang kehidupan dan ilmu pengetahuan [22].

2.2 Kata Baku

Kata baku adalah kata yang digunakan sesuai dengan aturan yang telah ditetapkan, seperti Pedoman Umum Ejaan Bahasa Indonesia (PUEBI), tata bahasa baku, dan Kamus Besar Bahasa Indonesia (KBBI). Ini memastikan konsistensi dan keseragaman dalam penggunaan bahasa Indonesia [22].

Kata baku merupakan istilah yang merujuk pada kata-kata dalam bahasa Indonesia yang telah disesuaikan dengan kaidah atau aturan yang telah ditetapkan. Kata-kata baku umumnya digunakan dalam komunikasi formal baik secara tertulis maupun lisan. Sebaliknya, kata tidak baku adalah kata yang digunakan secara tidak sesuai dengan aturan kebahasaan yang berlaku. Ketidaksesuaian tersebut dapat disebabkan oleh kesalahan dalam penulisan, pengucapan yang salah, atau susunan kalimat yang tidak tepat. Kata-kata tidak baku sering ditemui dalam tulisan, termasuk dalam karya sastra seperti cerpen. Penelitian ini bertujuan untuk mengidentifikasi kesalahan penggunaan kata baku dalam cerpen ”Warisan untuk Doni” karya Putu Ayub. Metode penelitian yang digunakan adalah pendekatan kualitatif deskriptif, di mana data yang dikumpulkan dan dianalisis merupakan

kata-kata dan kalimat dari cerpen tersebut. Metode pengumpulan data yang digunakan adalah dengan melakukan pembacaan dan pencatatan terhadap cerpen tersebut. Hasil analisis menunjukkan adanya berbagai kesalahan bahasa dalam cerpen tersebut, termasuk kesalahan penggunaan kata baku sebanyak sepuluh kasus, kesalahan penggunaan konjungsi sebanyak tiga kasus, dan kesalahan penggunaan tanda baca sebanyak tiga kasus. Penelitian ini memiliki manfaat sebagai sarana untuk menyampaikan gagasan peneliti dan sebagai sumber pembelajaran bagi pembaca mengenai kesalahan-kesalahan yang sering terjadi dalam penulisan bahasa Indonesia [23].

2.3 Natural Language Processing

Natural Language Processing adalah cabang dari kecerdasan buatan yang berfokus pada pemahaman dan produksi bahasa manusia alami, memungkinkan komputer berinteraksi dengan manusia tanpa menggunakan bahasa yang dikendalikan oleh komputer [24].

Naturan Language Processing memerlukan operasi pra-pemrosesan seperti tokenisasi, pemisahan, dan stemming dapat secara signifikan meningkatkan kualitas data kode sumber tak terstruktur untuk teknik pengambilan informasi [25]. Menurut Damerau, setelah dilakukan *pre-processing*, pemrosesan dapat dilanjutkan dengan berbagai teknik untuk memproses dan menganalisis bahasa manusia, termasuk segmentasi kalimat, analisis leksikal, analisis semantik, struktur wacana, dan pengenalan niat [26].

2.4 Text Preprocessing

Preprocessing teks merupakan serangkaian langkah penting yang dilakukan sebelum analisis dimulai, yang melibatkan identifikasi unit-unit seperti kata dan frasa yang akan digunakan, penghapusan konten yang tidak relevan seperti karakter nonalfabet dan kata-kata penghubung, penggabungan istilah-istilah yang memiliki hubungan semantis untuk mengurangi kejaran data dan meningkatkan kemampuan prediksi, serta peningkatan jumlah informasi semantis yang tersedia dengan menangani konsep negasi [27].

Tidak hanya itu, preprocessing teks juga memiliki dampak signifikan terhadap hasil aplikasi pemrosesan bahasa alami (NLP), di mana tokenisasi yang tepat dapat meningkatkan akurasi dalam penandaan bagian-bagian ucapan (POS),

sementara mempertahankan ekspresi multi-kata dapat meningkatkan penalaran dan terjemahan mesin. Oleh karena itu, korpus teks harus melalui proses preprocessing yang sesuai sebelum dapat digunakan sebagai input untuk model komputer. Persyaratan preprocessing ini bergantung pada sifat korpus serta tujuan dari aplikasi NLP itu sendiri yang ingin dicapai oleh para peneliti dalam menganalisis data [28]. Berikut merupakan proses yang umum dilakukan pada proses *preprocessing* teks dalam NLP.

1. Penghapusan Karakter Khusus dan Angka / Cleaning
   Menghapus karakter yang tidak relevan seperti tanda baca, angka, dan simbol khusus dapat membersihkan teks dari elemen yang tidak memberikan nilai informatif yang signifikan untuk analisis.

2. Lowercasing
   Mengonversi semua huruf dalam teks menjadi huruf kecil dengan tujuan mengurangi variasi yang tidak perlu, karena perbedaan huruf kapital dalam kata yang sama dan memiliki makna yang sama dapat berpengaruh dalam *text processing*.

3. Tokenisasi
   Memecah teks menjadi unit-unit terkecil yang disebut token (biasanya kata atau frasa) dengan tujuan memudahkan analisis teks dengan mengidentifikasi elemen dasar yang akan dianalisis.

4. Penghapusan Stop Words
   kata-kata umum yang tidak memberikan nilai berarti dalam analisis, seperti ”dan”, ”atau”, ”adalah” dengan tujuan mengurangi dimensi data dan fokus pada kata-kata yang lebih informatif.

5. Stemming
   Mengonversi kata-kata ke bentuk dasarnya dengan memotong akhiran (misalnya, ”berlari” menjadi ”lari”) dengan tujuan mengurangi variasi kata yang berasal dari kata dasar yang sama sehingga analisis lebih sederhana dan konsisten.

6. Lemmatization
   Mirip dengan stemming, tetapi menggunakan pendekatan linguistik untuk mengubah kata-kata ke bentuk dasar atau lema yang benar (misalnya,

”makan”, ”memakan”, dan ”dimakan” menjadi ”makan”) dengan tujuan memberikan bentuk dasar kata yang lebih akurat dibandingkan stemming.

7. Penghapusan Kata-kata Langka
   Menghapus kata-kata yang sangat jarang muncul dalam teks dengan tujuan mengurangi *noise* dalam data dan fokus pada kata-kata yang lebih signifikan.

8. Normalization
   Proses penyeragaman teks seperti mengubah singkatan menjadi bentuk panjangnya (misalnya, ”tdk” menjadi ”tidak”) dengan tujuan meningkatkan konsistensi dalam data teks.

9. POS Tagging
   Menandai setiap kata dengan jenis kata yang sesuai seperti kata benda, kata kerja, dan kata sifat dengan tujuan menyediakan konteks tambahan yang dapat membantu dalam analisis lebih lanjut, seperti dalam analisis sintaksis atau ekstraksi informasi.

10. Named Entity Recognition (NER)
    Mengidentifikasi dan mengklasifikasikan entitas bernama dalam teks seperti nama orang, organisasi, dan lokasi dengan tujuan mengekstraksi informasi spesifik yang dapat digunakan dalam analisis lanjutan seperti pencarian informasi atau pemahaman teks.

2.5 Levenshtein Distance

Algoritma Levenshtein Distance adalah teknik yang umum digunakan untuk mengukur kemiripan antara string, dengan berbagai peningkatan dan variasi yang telah dikembangkan untuk meningkatkan aplikasinya di berbagai bidang. Namun, efektivitasnya dapat bervariasi tergantung pada konteksnya, dan dalam beberapa kasus, pendekatan yang lebih canggih mungkin diperlukan [29].

Penyempurnaan pada algoritma dasar Levenshtein Distance telah diusulkan untuk menangani masalah-masalah tertentu, seperti pengambilan kosakata, perhitungan kesamaan string, dan pencarian basis data biologis. Penyempurnaan tersebut menghasilkan metode-metode yang ditingkatkan untuk mengurangi kompleksitas komputasional dan meningkatkan kinerja dalam tugas-tugas seperti pengecekan ejaan dan perhitungan kesamaan kalimat [29]. Perhitungan matematika algoritma Levenshtein Distance dapat dilihat pada persamaan 2.1-2.4.

$$D(s,t) = minD(s-1,t)+1 \quad \text{(penghapusan)} \tag{2.1}$$

$$D(s,t) = minD(s,t-1)+1 \quad \text{(penyisipan)} \tag{2.2}$$

$$D(s,t) = minD(s-1,t-1)+1,\ s_j \neq t_i \quad \text{(penggantian)} \tag{2.3}$$

$$D(s,t) = minD(s-1,t-1),\ s_j = t_i \quad \text{(tanpa perubahan)} \tag{2.4}$$

Keterangan :

1. s = String sumber
2. t = String target
3. D = Jarak edit Levenshtein Distance
4. $s(j)$ = Karakter string sumber ke-j
5. $t(i)$ = Karakter string target ke-i

Levenshtein Distance adalah metode yang diusulkan untuk mengukur tingkat kesalahan dalam tugas memasukkan teks, memungkinkan gaya interaksi yang lebih alami tanpa perlu bagi subjek untuk menjaga sinkronisasi dengan teks yang disajikan [30]. Metode Levenshtein Distance terbukti menjadi metode yang berhasil dan umum digunakan untuk mengukur jarak pada dua *string*.

2.6 Pustaka FuzzyWuzzy

Pustaka FuzzyWuzzy menggunakan Levenshtein Distance untuk menghitung perbedaan antara urutan karakter pada string. Pustaka FuzzyWuzzy menyediakan beberapa metode untuk membandingkan string, antara lain FuzzyWuzzy Ratio, FuzzyWuzzy PartialRatio, FuzzyWuzzy TokenSortRatio, FuzzyWuzzy TokenSetRatio, dan FuzzyWuzzy WRatio. Masingmasing metode memiliki kelebihan dan kekurangan dalam hal penggunaannya, tergantung pada kebutuhan pengguna. Dalam penggunaannya, pustaka FuzzyWuzzy akan mengembalikan skor yang berkisar dari 0 hingga 100. Skor tersebut menunjukkan seberapa mirip kedua string yang dibandingkan, dimana semakin tinggi skor menunjukkan semakin mirip kedua string tersebut [31].

Tabel 2.1. Tabel Research Gap

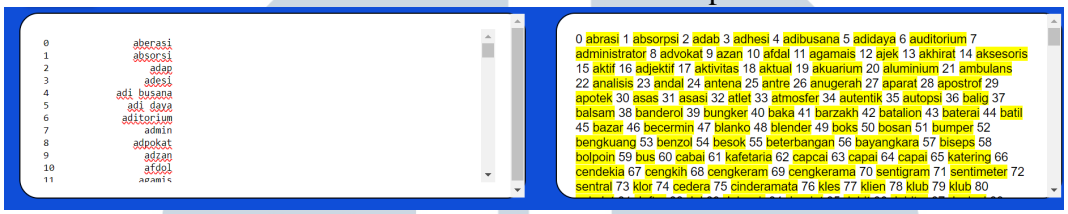
U-Tapis Kata Baku	**ejaan.id**	**typoonline**	**Gramatika**	**cek-ejaan.com**
Mampu memproses input lebih dari 10.000 karakter (1000 artikel) dalam waktu 12 menit	Tidak mampu memproses input lebih dari 10.000 karakter *(website not responding)*	Tidak mampu memproses input lebih dari 10.000 karakter *(website not responding)*	Tidak mampu memproses input lebih dari 10.000 karakter *(website not responding)*	Tidak mampu memproses input lebih dari 10.000 karakter *(website not responding)*
Mampu membedakan kutipan langsung dan tidak langsung	Tidak mampu membedakan kutipan langsung dan tidak langsung	Tidak mampu membedakan kutipan langsung dan tidak langsung	Tidak mampu membedakan kutipan langsung dan tidak langsung	Tidak mampu membedakan kutipan langsung dan tidak langsung
Mampu mendeteksi 100% dari kata tidak baku yang berada pada *dataset* penelitian	Mampu mendeteksi 94% dari kata tidak baku yang berada pada *dataset* penelitian	Mampu mendeteksi 50% kata tidak baku yang berada pada *dataset* penelitian	Mampu mendeteksi 90% dari kata tidak baku yang berada pada *dataset* penelitian	Mampu mendeteksi 78% kata tidak baku yang berada pada *dataset*
Mampu memberikan detail mengenai kata sebelum dan sesudah perubahan	Mampu memberikan detail mengenai kata sebelum dan sesudah perubahan	Tidak mampu memberikan detail perubahan kata, fungsinya hanya mendeteksi kata mana yang typo	Mampu memberikan detail mengenai kata sebelum dan sesudah perubahan	Tidak mampu memberikan detail perubahan kata, fungsinya hanya mendeteksi kata mana yang memiliki kesalahan ejaan

Sumber : Hasil Olahan Peneliti, 2024

Tabel 2.1 menunjukkan perbedaan penelitian dengan penelitian-penelitian yang telah dilakukan sebelumnya oleh peneliti lain. Beberapa diantaranya memiliki

perbedaan dari segi penampilan *output*, hasil deteksi, data kata yang termasuk kata tidak baku, dan perbedaan fungsi deteksi seperti pendeteksian kalimat langsung. Beberapa bukti perbedaan dapat dilihat pada gambar 2.1, 2.2, 2.3, 2.4, dan 2.5.

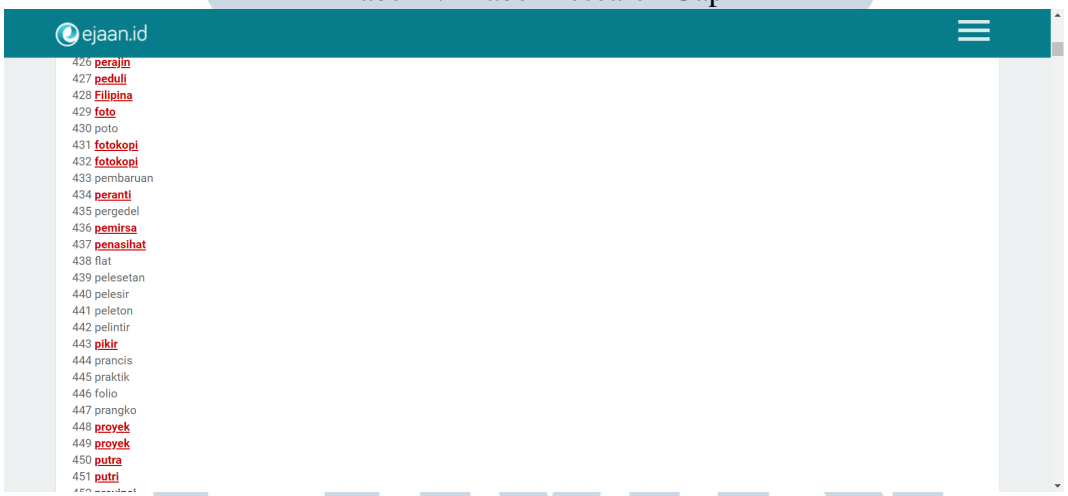
Tabel 2.1 Tabel Research Gap



Gambar 2.1. Hasil deteksi pada halaman situs penelitian U-Tapis kata baku

Sumber : Hasil Olahan Peneliti pada Aplikasi Deteksi dan Perbaikan Kesalahan Kata Baku U-Tapis, 2024

Tabel 2.1 Tabel Research Gap



Gambar 2.2. Hasil deteksi pada halaman situs ejaan.id

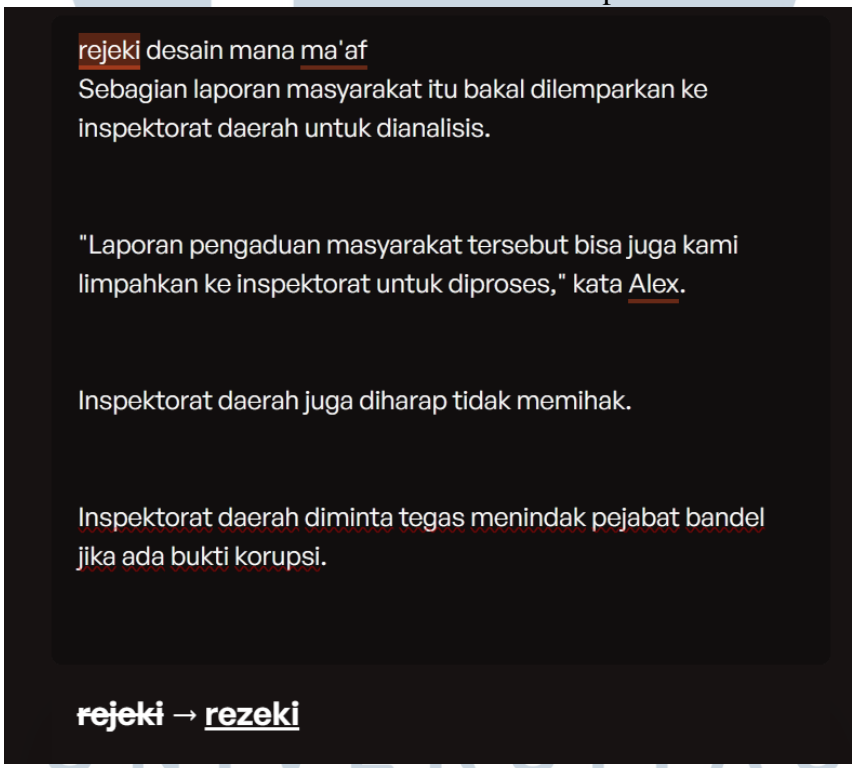
Sumber : Hasil Olahan Peneliti pada Aplikasi Ejaan.id, 2024

Tabel 2.1 Tabel Research Gap

Gambar 2.3. Hasil deteksi pada halaman situs typoonline

Sumber : Hasil Olahan Peneliti pada Aplikasi Typoonline, 2024

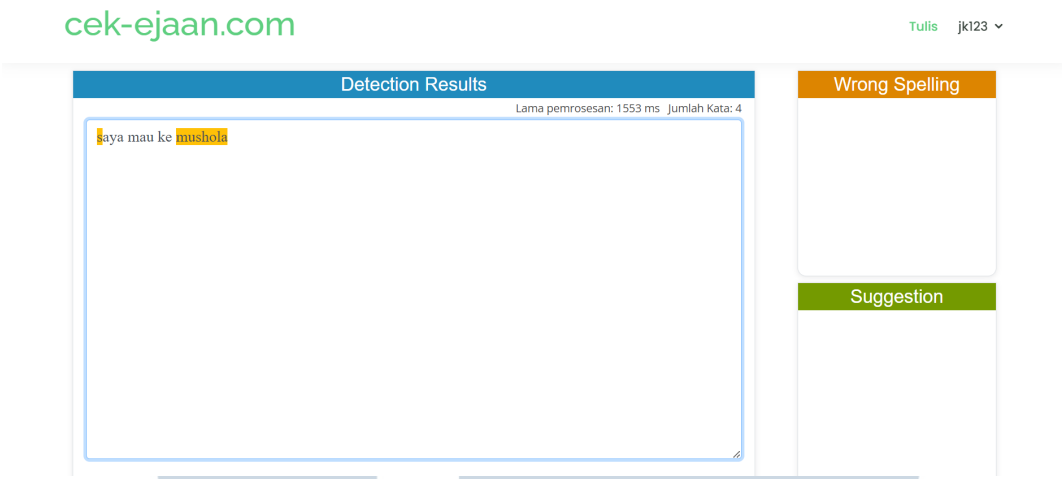
Tabel 2.1 Tabel Research Gap



Gambar 2.4. Hasil deteksi pada halaman situs Gramatika

Sumber : Hasil Olahan Peneliti pada Aplikasi Gramatika, 2024

Tabel 2.1 Tabel Research Gap



Gambar 2.5. Hasil deteksi pada halaman situs cek-ejaan.com

Sumber : Hasil Olahan Peneliti pada Aplikasi Cek-Ejaan.com, 2024