BAB 2
LANDASAN TEORI

2.1 Kesalahan Ejaan

Kesalahan ejaan dapat dibagi menjadi dua, yaitu kesalahan ejaan *non-word* dan kesalahan ejaan *real-word* [8].

1. Kesalahan ejaan *non-word*

   Kesalahan ejaan *non-word* merupakan kesalahan ejaan kata dimana kata yang terdapat kesalahan tidak ada di dalam kamus atau bisa dikatakan tidak memiliki arti. Contoh : "Para **ibu** rumah tangga" menjadi "Para **inu** rumah tangga". Kata "inu" merupakan kesalahan kata *real-word*, karena kata tersebut tidak terdapat di dalam kamus. Algoritma untuk kesalahan *non-word* telah banyak diteliti seperti algoritma *spell-checker* atau *typo-checker*, dimana algoritma ini diaplikasikan pada banyak perangkat lunak pengolahan kata, contohnya fitur *auto-correct* pada *keyboard*.

2. Kesalahan ejaan *real-word*

   Kesalahan ejaan *real-word* merupakan kesalahan ejaan kata dimana kata yang terdapat kesalahan ada di dalam kamus, namun jika dilihat secara konteks menjadi tidak sesuai karena bukan merupakan kata yang dimaksud. Contoh : "Para **ibu** rumah tangga" menjadi "Para **itu** rumah tangga". Kata "itu" adalah kesalahan kata *real-word* karena kata itu terdapat dalam kamus tetapi bukan merupakan kata yang dimaksud.

Algoritma *typo-checker* yang dibuat untuk mengatasi kesalahan ejaan *non-word* tidak dapat menangani kesalahan ejaan *real-word* karena terbatas pada penanganan kata yang tidak terdapat dalam kamus. Berdasarkan penelitian yang dilakukan oleh Kukich, kesalahan ejaan *real-word* mencakup sekitar 25%-40% dari seluruh kesalahan ejaan kata yang telah terdokumentasi [9].

2.2 Penelitian Terdahulu

Kesalahan ejaan *real-word* secara umum memiliki dua pendekatan pada sistem deteksi dan koreksi kesalahan kata yaitu pendekatan yang berbasis sumber

daya dan pendekatan berbasis *machine learning* dan statistik. Pendekatan berbasis sumber daya adalah pendekatan yang menggunakan sumber daya leksikal untuk mengetahui konteks dari kalimat yang diperiksa apakah sudah sesuai. Salah satu penelitian yang menggunakan pendeketan berbasis sumber daya leksikal adalah Hirst dan Budanitsky [8] dengan memeriksa jarak semantik kalimat yaitu dengan mengidentifikasi token-token kemudian menggantinya dengan varian ejaan dari kata yang berhubungan dengan konteks kalimat tersebut.

Pendekatan berbasis *machine learning* dan statistik adalah pendeketan yang menggunakan pemodelan bahasa statistik yang memperkirakan kemungkinan suatu kata muncul di dalam konteks kalimat lain melalui tabel-tabel perkiraan. Pada umumnya pendekatan berbasis *machine learning* dan statistik ini bergantung pada *confusion set* yang dibentuk, yaitu himpunan kata-kata yang biasanya tertukar satu sama lain. Contoh *confusion set* untuk kata “ibu” = { ibu, itu, isu }.

Salah satu metode dengan pendekatan *machine learning* dan statistik adalah metode Demerau dan Mercer (MDM) yang diajukan oleh Mays,. Metode ini melihat nilai probabilitas dari trigram-trigram kata yang terbentuk. Jika perubahan salah satu kata dengan varian ejaan menghasilkan nilai yang lebih kecil dibandingkan dengan tidak mengubahnya, maka dapat dibuat hipotesis bahwa kata asli salah dan kata variasinya benar. Dengan kata lain, kecilnya nilai probabilitas trigram kata menandakan terdapat kesalahan *real-word* [10].

O’Hearn membandingkan metode MDM dengan metode dari Hirst dan Budanitsky yang menggunakan pendeketan sumber daya leksikal dan ditemukan bahwa metode MDM lebih baik dengan menggunakan data uji yang sama. Metode MDM menghasilkan precision 54%-79% dan recall 25%-64% sedangkan 9 metode Hirst dan Budanitsky [8] menghasilkan precision 18%-25% dan recall 23%-50%.

Ken Wite Ariing Cahyu telah melakukan penelitian terkait kesalahan ejaan *real-word* Bahasa Indonesia dengan menggunakan metode *kneser-ney smoothing*. Didapatkan hasil akurasi deteksi dan koreksi sebesar 19,8% dan 95,5% dengan parameter D paling optimal yaitu 0,0001. Pada penelitian ini akan

mengembangkan penelitian Ken dengan menggunakan metode *modified kneser-ney smoothing* yang merupakan modifikasi dari penelitian sebelumnya dimana dikatakan memiliki performa yang sangat baik dengan menggunakan 3 tingkatan nilai diskon yang dikembangkan oleh Stanley dan Goodman.

2.3 Deteksi dan Koreksi Kesalahan *Real-Word*

Metode bigram dan trigram ini melakukan pendeteksian dan pengoreksian kesalahan dengan melihat masing-masing bigram dan trigram kata kandidat. Selanjutnya akan dilakukan perhitungan skor untuk setiap *confusion set* sehingga akan diberikan sugesti koreksi kata.

2.3.1 *Confusion Set* dengan *Levenshtein Distance*

Confusion set merupakan kumpulan kata yang biasanya tertukar. Dalam membuat *confusion set* digunakan perhitungan *Levenshtein Distance* [11] untuk menghitung jarak minimum suatu kata (*minimum edit distance*), dimana jumlah operasi *edit* yang dibutuhkan untuk bisa mengubah kata menjadi kata yang lain. Operasi editnya adalah pemasukan (*insertion*), pergantian (*substitution*), dan penghapusan (*deletion*). Berikut adalah perhitungan *Levenshtein Distance*.

$$d(\mathrm{i,j}) = \min \begin{cases} d(i-1,j) + 1 \\ d(i,j-1) + 1 \\ d\{i-1, j-1 + cost \end{cases}$$

Dengan i adalah baris dan j adalah kolom. Hasil perhitungan didapatkan pada hasil perhitungan di baris dan kolom terakhir matriks. Kata-kata yang dijadikan daftar *confusion set* adalah kata yang mempunya nilai *Levenshtein Distance* maksimal 1. *Confusion set* dapat direpresentasikan sebagai berikut.

$$C(W^i) = \{c_1^i, c_2^i, \dots, c_j^i, c_{k_i}^i\}$$

Dimana W^i adalah kata ke-i di dalam kalimat dan k_i adalah jumlah elemen yang terdapat dalam $C(W^i)$.

2.3.2 Membuat Model N-Gram

N-Gram adalah urutan kata yang didapatkan dari teks. Biasanya n-gram yang digunakan adalah unigram, bigram dan trigram. Dengan nilai n dari unigram

adalah 1, nilai n dari bigram adalah 2 dan nilai n dari trigram adalah 3. Contoh n-gram untuk kalimat “para ibu rumah tangga”:

1-gram(unigram) = “para”, “ibu”, “rumah”, “tangga”

2-gram(bigram) = “_para”, “para ibu”, “ibu rumah”, “rumah tangga”, “tangga_”

3-gram(trigram) = “_para ibu”, “para ibu rumah”, “ibu rumah tangga”, “rumah tangga_”

Model N-Gram dibuat dengan membangkitkan himpunan bigram dan trigram [10] untuk setiap anggota *confusion set*, dimana bigram terdiri dari bigram kiri dan bigram kanan. Kata yang diambil untuk bigram dan trigram yaitu kata sebelumnya dan sesudahnya. Kata ke-i dalam anggota *confusion set* dilambangkan dengan c_j^i. Maka unigram, bigram dan trigram yang terbentuk adalah:

Unigram : c_j^i

Bigram kiri : $W^{i-1}c_j^i$

Bigram kanan : $c_j^i W^{i-1}$

Trigram : $W^{i-1}c_j^i W^{i+1}$

2.3.3 Perhitungan Probabilitas N-Gram

Probabilitas N-Gram dihitung menggunakan Maximum Likehood Estimation (MLE) yaitu dengan mengambil asumsi bahwa kemunculan kata bergantung pada kemunculan kata sebelumnya dan sesudahnya dalam suatu kalimat. Dengan demikian didapatkan probabilitas unigram, bigram dan trigram sebagai berikut.

Probabilitas dari kata c_j^i

$$P_0(c_j^i) = \frac{count(c_j^i)}{\sum_r^{k_i} count(c_r^i)}$$

$$P_1(c_j^i|W^{i-1}) = \frac{count(W^{i-1}c_j^i)}{\sum_r^{k_i} count(W^{i-1}c_r^i)}$$

$$P_2(c_j^i|W^{i+1}) = \frac{count(c_j^i W^{i+1})}{\sum_r^{k_i} count(c_r^i W^{i+1})}$$

$$P_3(c_j^i|W^{i-1},W^{i+1}) = \frac{count(W^{i-1}c_j^iW^{i+1})}{\sum_r^{k_i} count(W^{i-1}c_r^iW^{i+1})}$$

Dimana P_1 merupakan probabilitas untuk setiap elemen *confusion set* dan satu kata sebelumnya untuk bigram kiri, sedangkan P_2 merupakan probabilitas untuk setiap elemen *confusion set* dan satu kata setelahnya untuk bigram kanan dan P_3 merupakan probabilitas untuk setiap elemen *confusion set* dan satu kata sebelumnya dan sesudahnya untuk trigram. Namun dengan menggunakan rumus diatas terdapat hasil probabilitas yang bernilai nol jika tidak terdapat kemunculan kata, sehingga muncul metode-metode *smoothing* yang menambahkan pseudocount pada probabilitas yang bernilai nol. Metode *smoothing modified Kneser-Ney* menghitung probabilitas menggunakan kemunculan kata pada *confusion set* yang lebih dari nol dan menggunakan nilai diskon yang bertingkat sesuai dengan banyaknya kemunculan. Berikut merupakan contoh perhitungan probabilitas menggunakan metode *modified kneser-ney smoothing*. Maka rumus-rumus probabilitas n-gram menjadi:

Rumus dasar : [4]

$$P_{KN}(w_i|w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^i) - D\left(c(w_{i-n+1}^i)\right)}{\sum_{w_i} c(w_{i-n+1}^i)} + \gamma(w_{i-n+1}^{i-1})\, P_{KN}(w_i|w_{i-n+2}^{i-1})$$

Dengan penjabaran untuk bigram dan trigram sebagai berikut :

1. Bigram

   Bigram kiri

   Perhitungan metode *smoothing modified kneser-ney* untuk bigram kiri menggunakan persamaan :

$$P_1(c_j^i|W^{i-1}) = \frac{count(W^{i-1}c_j^i) - D\left(count(W^{i-1}c_j^i)\right)}{\sum_{r=1}^{k_i} count(W^{i-1}c_r^i)} + \gamma(c_j^i|W^{i-1}) \times \frac{N_{1+}(c_j^i)}{\sum_{r=1}^{k_i} N_{1+}(c_r^i)}$$

   Untuk membuat jumlah distribusi menjadi 1, maka :

$$\gamma(c_j^i|W^{i-1}) = \frac{D_1N_1(c_j^i|W^{i-1}) + D_2N_2(c_j^i|W^{i-1}) + D_{3+}N_{3+}(c_j^i|W^{i-1})}{\sum_{r=1}^{k_i} count(W^{i-1}c_r^i)}$$

Dimana nilai $D\left(count\left(W^{i-1}c_j^i\right)\right)$ adalah :

$$\mathrm{D(count)} = \begin{cases} 0, & jika\ c = 0 \\ D_1, & jika\ c = 1 \\ D_2, & jika\ c = 2 \\ D_{3+}, & jika\ c \geq 3 \end{cases}$$

Nilai D (*discount*) dihubungkan dengan analogi :

$$Y = \frac{n_1}{n_1 + 2n_2}$$

$$D_1 = 1 - 2Y\ \frac{n_2}{n_1}$$

$$D_2 = 2 - 3Y\ \frac{n_3}{n_2}$$

$$D_{3+} = 3 - 4Y\ \frac{n_4}{n_3}$$

Dimana, n_1 dan n_2 adalah jumlah total bigram kiri dengan masing-masing nilai hitungan tepat satu untuk n_1 dan tepat dua untuk n_2, dan seterusnya.

$$N_1\left(c_j^i \middle| W^{i-1}\right) = \left|\left\{W^i : count\left(W^{i-1}c_j^i\right) = 1\right\}\right|$$

$N_1\left(c_j^i \middle| W^{i-1}\right)$ merupakan notasi dari nilai kata W^i dimana jumlah kemunculan bigram kiri dalam anggota *confusion set* sama dengan 1

$$N_2\left(c_j^i \middle| W^{i-1}\right) = \left|\left\{W^i : count\left(W^{i-1}c_j^i\right) = 2\right\}\right|$$

$N_2\left(c_j^i \middle| W^{i-1}\right)$ merupakan notasi dari nilai kata W^i dimana jumlah kemunculan bigram kiri dalam anggota *confusion set* sama dengan 2

$$N_{3+}(c_j^i | W^{i-1}) = \left|\left\{W^i : count\left(W^{i-1}c_j^i\right) > 2\right\}\right|$$

$N_{3+}\left(c_j^i \middle| W^{i-1}\right)$ merupakan notasi dari nilai kata W^i dimana jumlah kemunculan bigram kiri dalam anggota *confusion set* yang lebih dari 2

$$N_{1+}\left(c_j^i\right) = \left|\left\{W^i : count\left(c_j^i\right) > 0\right\}\right|$$

$N_{1+}\left(c_j^i\right)$ merupakan notasi dari nilai kata W^i dimana jumlah kemunculan kata unigram dalam anggota *confusion set* yang lebih dari 0.

$\sum_{r=1}^{k_i} N_{1+}\left(c_r^i\right)$ merupakan notasi dari total jumlah kemunculan semua anggota *confusion set* dari unigram yang lebih dari 0.

Bigram kanan

Perhitungan metode *smoothing modified kneser-ney* untuk bigram kanan menggunakan persamaan :

$$P_2(c_j^i|W^{i+1}) = \frac{count(c_j^i W^{i+1}) - D\left(count(c_j^i W^{i+1})\right)}{\sum_{r=1}^{k_i} count(c_r^i W^{i+1})} + \gamma(c_j^i|W^{i+1}) \times \frac{N_{1+}(c_j^i)}{\sum_{r=1}^{k_i} N_{1+}(c_r^i)}$$

Untuk membuat jumlah distribusi menjadi 1, maka :

$$\gamma(c_j^i|W^{i+1}) = \frac{D_1 N_1(c_j^i|W^{i+1}) + D_2 N_2(c_j^i|W^{i+1}) + D_{3+} N_{3+}(c_j^i|W^{i+1})}{\sum_{r=1}^{k_i} count(c_r^i W^{i+1})}$$

Dimana nilai $D\left(count(c_j^i W^{i+1})\right)$ adalah :

$$\mathrm{D(count)} = \begin{cases} 0, & jika\ c = 0 \\ D_1, & jika\ c = 1 \\ D_2, & jika\ c = 2 \\ D_{3+}, & jika\ c \geq 3 \end{cases}$$

Nilai D (*discount*) dihubungkan dengan analogi :

$$Y = \frac{n_1}{n_1 + 2n_2}$$

$$D_1 = 1 - 2Y\,\frac{n_2}{n_1}$$

$$D_2 = 2 - 3Y\,\frac{n_3}{n_2}$$

$$D_{3+} = 3 - 4Y\,\frac{n_4}{n_3}$$

Dimana, n_1 dan n_2 adalah jumlah total bigram kanan dengan masing-masing nilai hitungan tepat satu untuk n_1 dan tepat dua untuk n_2, dan seterusnya.

$$N_1(c_j^i|W^{i+1}) = |\{W^i : count(c_j^i W^{i+1}) = 1\}|$$

$N_1(c_j^i|W^{i+1})$ merupakan notasi dari nilai kata W^i dimana jumlah kemunculan bigram kanan dalam anggota *confusion set* sama dengan 1

$$N_2(c_j^i|W^{i+1}) = |\{W^i : count(c_j^i W^{i+1}) = 2\}|$$

$N_2(c_j^i|W^{i+1})$ merupakan notasi dari nilai kata W^i dimana jumlah kemunculan bigram kanan dalam anggota *confusion set* sama dengan 2

$$N_{3+}(c_j^i|W^{i+1}) = |\{W^i : count(c_j^i W^{i+1}) > 2\}|$$

$N_{3+}(c_j^i|W^{i+1})$ merupakan notasi dari nilai kata W^i dimana jumlah kemunculan bigram kanan dalam anggota *confusion set* yang lebih dari 2

$$N_{1+}(c_j^i) = |\{W^i : count(c_j^i) > 0\}|$$

$N_{1+}(c_j^i)$ merupakan notasi dari nilai kata W^i dimana jumlah kemunculan kata unigram dalam anggota *confusion set* yang lebih dari 0.

$\sum_{r=1}^{k_i} N_{1+}(c_r^i)$ merupakan notasi dari total jumlah kemunculan semua anggota *confusion set* dari unigram yang lebih dari 0.

2. Trigram

Perhitungan metode *Smoothing modified Kneser-Ney* untuk trigram menggunakan persamaan :

$$P_3(c_j^i|W^{i-1}, W^{i+1}) = \frac{count(W^{i-1}c_j^i W^{i+1}) - D\left(count(W^{i-1}c_j^i W^{i+1})\right)}{\sum_{r=1}^{k_i} count(W^{i-1}c_r^i W^{i+1})} + \gamma(c_j^i|W^{i-1}, W^{i+1}) \times \frac{\left(\frac{N_{1+}(c_j^i|W^{i-1})}{\sum_{r=1}^{k_i} N_{1+}(c_j^i|W^{i-1})}\right) + \left(\frac{N_{1+}(c_j^i|W^{i+1})}{\sum_{r=1}^{k_i} N_{1+}(c_j^i|W^{i+1})}\right)}{2}$$

Untuk membuat jumlah dristribusi menjadi 1, maka :

$$\gamma(c_j^i|W^{i-1}, W^{i+1}) = \frac{D_1 N_1(c_j^i|W^{i-1}, W^{i+1}) + D_2 N_2(c_j^i|W^{i-1}, W^{i+1}) + D_{3+} N_{3+}(c_j^i|W^{i-1}, W^{i+1})}{\sum_{r=1}^{k_i} count(W^{i-1}c_r^i W^{i+1})}$$

Dimana nilai $D\left(count(W^{i-1}c_j^i W^{i+1})\right)$ dapat dilihat pada lembar berikuntya:

$$\mathrm{D(count)} = \begin{cases} 0, & jika\ c = 0 \\ D_1, & jika\ c = 1 \\ D_2, & jika\ c = 2 \\ D_{3+}, & jika\ c \geq 3 \end{cases}$$

Nilai D (*discount*) dihubungkan dengan analogi :

$$Y = \frac{n_1}{n_1 + 2n_2}$$

$$D_1 = 1 - 2Y\ \frac{n_2}{n_1}$$

$$D_2 = 2 - 3Y\ \frac{n_3}{n_2}$$

$$D_{3+} = 3 - 4Y\ \frac{n_4}{n_3}$$

Dimana, n_1 dan n_2 adalah jumlah total trigram dengan masing-masing nilai hitungan tepat satu untuk n_1 dan tepat dua untuk n_2, dan seterusnya.

$$N_1\left(c_j^i \middle| W^{i-1}, W^{i+1}\right) = \left|\left\{W^i : count\left(W^{i-1} c_j^i W^{i+1}\right) = 1\right\}\right|$$

$N_1\left(c_j^i \middle| W^{i-1}, W^{i+1}\right)$ merupakan notasi dari nilai kata W^i dimana jumlah kemunculan trigram dalam anggota *confusion set* sama dengan 1

$$N_2\left(c_j^i \middle| W^{i-1}, W^{i+1}\right) = \left|\left\{W^i : count\left(W^{i-1} c_j^i W^{i+1}\right) = 2\right\}\right|$$

$N_2\left(c_j^i \middle| W^{i-1}, W^{i+1}\right)$ merupakan notasi dari nilai kata W^i dimana jumlah kemunculan trigram dalam anggota *confusion set* sama dengan 2

$$N_{3+}(c_j^i | W^{i-1}, W^{i+1}) = \left|\left\{W^i : count\left(W^{i-1} c_j^i W^{i+1}\right) > 2\right\}\right|$$

$N_{3+}\left(c_j^i \middle| W^{i-1}, W^{i+1}\right)$ merupakan notasi dari nilai kata W^i dimana jumlah kemunculan trigram dalam anggota *confusion set* yang lebih dari 2

$$\frac{\left(\frac{N_{1+}(c_j^i | W^{i-1})}{\sum_{r=1}^{k_i} N_{1+}(c_j^i | W^{i-1})}\right) + \left(\frac{N_{1+}(c_j^i | W^{i+1})}{\sum_{r=1}^{k_i} N_{1+}(c_j^i | W^{i+1})}\right)}{2}$$

Merupakan nilai rata-rata dari jumlah kemunculan bigram kiri dan kanan dalam anggota *confusion set* yang lebih dari 0 dibagi dengan total jumlah kemunculan semua anggota *confusion set* dari bigram kiri dan kanan yang lebih dari 0.

2.3.4 *Weighted Combination Score*

Model-model n-gram berorde tinggi maupun rendah memiliki kelebihan dan kekurangan. N-gram berorde lebih tinggi lebih sensitif terhadap konteks, namun memiliki jumlah kemunculan lebih kecil sedangkan sebaliknya n-gram berorde lebih rendah lebih terbatas mengenal konteks, namun memiliki jumlah kumunculan lebih besar. Oleh karena itu model ini menggabungkan bigram kiri dan kanan serta trigram agar tidak terlalu bergantung pada salah satu n-gram [10].

Maka persamaan untuk perhitungan *wighted combination score* probabilitas n-gram:

$$Score(W_j^i) = \lambda_1 P_1(W_j^i|W^{i-1}) + \lambda_2 P_2(W_j^i|W^{i+1}) + \lambda_3 P_3(W_j^i|W^{i-1}, W^{i+1})$$

Dengan nilai bobot λ_1 λ_2 λ_3 untuk setiap perhitungan probabilitas bigram kiri, bigram kanan dan trigram berdasarkan penelitian yang dilakukan oleh Samantha dengan nilai terbaik untuk setiap bobot adalah $\lambda_1 = \lambda_2 = 0.25$ dan $\lambda_3 = 0.5$.

2.3.5 Deteksi Kesalahan dan Pemilihan Kata Sugesti

Untuk mengetahui suatu kata merupakan kesalahan *real-word*, diberikan beberapa aturan. Pertama, elemen-elemen *confusion set* diurutkan mulai dari skor tertinggi sampai terendah. Dalam penelitian Mays diketahui nilai optimum yaitu 0.99 [12] dengan diyakini bahwa kata yang diuji dapat menjadi kesalahan kata *real-word* dalam 1% kasus. Maka dikatakan suatu kata adalah sebuah kesalahan *real-word* yaitu jika nilai skor kata yang diuji kurang dari 1% nilai skor tertinggi elemen-elemen *confusion set* yang telah diurutkan.

2.3.6 Rencana Pengujian

Kinerja dari metode *smoothing* yang digunakan dalam sistem ini diukur dengan melakukan pengukuran akurasi deteksi dan koreksi [2]. Pengukuran akurasi deteksi pada penelitian sebelumnya dilakukan dengan persamaan:

$$\text{Akurasi deteksi} = \frac{ds}{dm} \times 100\%$$

Dimana, ds : jumlah deteksi yang dihasilkan oleh sistem

dm : jumlah deteksi yang dihasilkan secara manual

Pada penelitian ini pengukuran akurasi deteksi menggunakan *confusin matrix* [13]. Dengan penentuan TP (True Positive), TN (True Negative), FP (False Positive), dan FN (False Negative) didasari pada benar salahnya sistem dalam mendeteksi atau mengelompokkan sebuah kata ke dalam kategori kata benar atau kata salah.

$$\text{Akurasi deteksi} = \frac{TP + TN}{TP + FP + FN + TN} \times 100\%$$

Sedangkan pengukuran akurasi koreksi dilakukan dengan persamaan:

$$\text{Akurasi Koreksi} = \frac{kb}{kb + ks} \times 100\%$$

Dimana, kb : jumlah koreksi benar yang dihasilkan oleh sistem

ks : jumlah koreksi salah yang dihasilkan oleh sistem

Pada penelitian ini pengukuran akurasi koreksi menggunakan *precission* [13]. Dengan penentuan TP (True Positive), dan FP (False Positive) didasari pada kata yang sebenarnya adalah salah namun dideteksi benar oleh sistem.

$$\text{Akurasi koreksi} = \frac{TP}{TP + FP} \times 100\%$$

2.4 Pemodelan

Pemodelan merupakan metode yang digunakan untuk memodelkan atau menggambarkan hubungan data yang ada di dalam sistem deteksi dan koreksi kesalahan kata yang akan dibangun. Pada penelitian ini diagram yang digunakan untuk pemodelan adalah Flowchart, Diagram Konteks, dan DFD (Data Flow Diagram).

2.4.1 Flowchart

Flowchart adalah bagan-bagan yang menjelaskan alur proses dari sebuah program. Dengan menggunakan flowchart akan memudahkan pengecekan bagian-bagian yang terlupakan dalam analisis masalah [14]. Tujuan membuat Flowchart adalah mempermudah penyelesaian masalah secara sederhana, terurai, rapi dan jelas menggunakan simbol-simbol standar.

2.4.2 Diagram Konteks

Diagram Konteks adalah diagram sederhana yang terdiri dari suatu proses dan menggambarkan hubungan antara entity luar, masukan dan keluaran dari sistem. [15]. Diagram konteks berfungsi menggambarkan transformasi dari suatu proses yang melakukan transformasi data input dan menjadi transformasi data output. Diagram konteks menggaris bawahi sejumlah karakteristik penting dari suatu sistem yaitu :

1. Menggunakan hanya satu simbol proses.
2. Memberi label simbol proses tersebut untuk menggambar seluruh sistem, biasanya berupa kata kerja di tambah objek.
3. Tidak memberi nomor pada simbol proses.
4. Menyertakan semua terminator dari system.
5. Menunjukan semua arus data antara terminator dan system.

2.4.3 DFD (Data Flow Diagram)

DFD adalah suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut [16]. Berikut merupakan komponen dari DFD.

1. Arus Data (Data Flow)

   Menunjukan arus dari data yang dapat berupa masukan untuk sistem atau dari proses sistem.
2. Proses

   Proses adalah kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.
3. Kesatuan Luar (External Entity)

   Kesatuan luar merupakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lain yang akan memberikan masukan atau menerima keluaran dari sistem.

4. File

   Kumpulan data yang disimpan dengan cara tertentu. Data yang mengalir disimpan dalam file. Aliran data di-update atau ditambahkan ke dalam file.

2.4.4 Kamus Data

Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (input) dan keluaran (output) dapat dipahami secara umum [16]. Kamus data mendefinisikan elemen data dengan menjelaskan arti aliran data dan penyimpanan data dalam DFD, mendeskripsikan komposisi paket data yang bergerak melalui aliran, mendeskripsikan komposisi penyimpanan data, menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran, mendeskripsikan hubungan detil antar penyimpanan.

Kamus data berfungsi untuk membantu pelaku sistem untuk mengartikan alokasi secara detail dan mengorganisasikan semua elemen data yang digunakan dalam sistem secara persis sehingga baik pemakai atau penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses

Kamus data biasanya berisi:

1. Nama – nama dari data
2. Digunakan pada – merupakan proses-proses yang terkait data
3. Deskripsi – merupakan deskripsi data
4. Informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data

Kamus data memiliki simbol untuk menjelaskan informasi tambahan seperti pada tabel 2.1 berikut.

Tabel 2.1 Kamus Data

Simbol	Keterangan
=	Disusun atau terdiri dari
+	Dan
[|]	baik ... atau ...
{}n	n kali diulang/bernilai banyaj

()	data opsional
...	batas komentar

2.5 Bahasa Pemrograman

Bahasa pemrograman merupakan kumpulan aturan sintaks dan semantik yang berfungsi untuk mendefinisikan program komputer untuk dapat dijalankan. Bahasa pemrograman memerintah komputer untuk mengolah data sesuai dengan alur berpikir yang manusia perintahkan[17]. Beberapa bahasa pemrograman yang banyak digunakan antara lain Java, JavaScript, PHP, C, C++ dan Python. Pada pembangunan sistem deteksi dan koreksi kesalahan kata ini, bahasa pemrograman yang digunakan adalah PHP dan JavaScript.

2.5.1 PHP

PHP atau (Hypertext Preprocessor) merupakan bahasa pemrograman yang berada pada sisi server (script server-side) yang didesain untuk pengembangan website. Disebut script server-side karena diproses pada komputer server[18].

Kelebihan menggunakan PHP adalah tidak diedarkannya kode sumber yang membangun PHP ke sisi klien sehingga kerahasiaan kode dapat dilindungi. PHP berjalan pada sisi server, sehingga untuk dapat menggunakannya kita harus mengaktifkan web server terlebih dahulu baik offline maupun online.

2.5.2 Javascript

Javascript merupakan bahasa pemrograman yang berbentuk script yang berfungsi untuk memberikan tampilan yang kelihatan lebih interaktif pada website yang dibangun. Bahasa pemrograman ini memberikan kemampuan tambahan kepada HTML(Hypertext Markup Language) dengan melakukan eksekusi perintah pada sisi client (script client-side)[18]. Meskipun secara umum digunakan pada script client-side namun dapat pula digunakan pada script server-side dengan memprogramnya sebagai bahasa python atau perl[19].

Javascript berfungsi untuk mendeteksi dan merespon event-event yang diberikan oleh pengguna. Javascript dapat digunakan untuk perhitungan

aritmatika, manipulasi tanggal dan waktu, modifikasi array dan menangani event yang diinisiasi oleh pengguna serta menetapkan waktu.

2.6 *Database*

Database merupakan suatu komponen yang penting dalam sebuah sistem karena di dalam *database* tersimpan semua informasi yang akan diolah dan dihasilkan akan tersimpan di dalam *database*[20]. *Database* merupakan kumpulan file-file yang berhubungan secara logis dan digunakan secara rutin pada operasi-operasi sistem informasi manajemen. Semua *database* umumnya berisi elemen-elemen data yang disusun ke dalam file-file yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, maka *database* menunjukkan suatu kumpulan tabel yang dipakai dalam suatu perusahaan atau instansi untuk tujuan tertentu[21].

Dalam pengelolaan *database* digunakan sebuah sistem manajemen *database* yaitu *database* management system (DBMS). Software yang digunakan untuk mengelola *database* pada pembangunan sistem deteksi dan koreksi kesalahan kata menggunakan MySQL dengan bahasa SQL. Berikut penjelasannya.

2.6.1 SQL (Structured Query Language)

Structured Query Language (SQL) adalah bahasa yang digunakan untuk mengelola manajemen data pada relational *database* management systems (RDMS). SQL adalah bahasa standar komputer yang pada awalnya dikembangkan oleh IBM untuk query mengubah dan mendefinisikan basis data relasional, menggunakan pernyataan deklaratif[22].

Terdapat 3 jenis perintah SQL[23], yaitu dapat dilihat pada halaman berikutnya :

1. Data Control Language (DCL)

   DCL merupakan perintah SQL yang berhubungan dengan pengelolaan user dan hak akses terhadap setiap objek di MySQL. Perintah SQL yang termasuk di dalam DCL adalah GRANT, REVOKE.

2. Data Definition Language (DDL)

   DDL merupakan perintah SQL yang berhubungan dengan definisi dari suatu struktur *database*, yaitu *database* dan table. Perintah dasar yang termasuk DDL adalah CREATE, ALTER, RENAME, DROP, SHOW.

3. Data Manipulation Language (DML)

   DML merupakan perintah SQL yang berhubungan dengan manipulasi atau pengolahan data atau record dalam table. Perintah yang termasuk di dalam DML adalah SELECT, INSERT, UPDATE, DELETE.

2.6.2 MySQL

MySQL merupakan *database* server yang bersifat multiuser dan multithreaded. SQL adalah bahasa *database* standar yang memudahkan penyimpanan, pengubahan dan akses informasi[24]. Pada MySQL dikenal dengan istilah *database* dan tabel. Tabel sendiri merupakan sebuah struktur data dua dimensi yang terdiri dari baris-baris record dan kolom. MySQL termasuk salah satu *database* open source yang paling banyak digunakan karena selain gratis MySQL pun mempunyai banyak dukungan bahasa pemrograman dan aplikasi sebagai solusi *database*. Dalam penelitian ini MySQL digunakan untuk membuat dan mengolah *database* korpus kamus dan korpus n-gram beserta isinya.

2.7 Perangkat Lunak Pengembangan Sistem

Perangkat lunak adalah istilah khusus untuk data yang diformat, dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca dan ditulis oleh komputer. Dengan kata lain, bagian sistem komputer yang tidak berwujud.

Dalam pembangunan dan pengujian sistem koreksi kesalahan ejaan *real-word* digunakan XAMPP sebagai *web server* sedangkan untuk pembangunan dan pengujian sistem deteksi dan koreksi kesalahan ejaan *real-word* yang dibangun menggunakan *web browser* dan yang terakhir bahasa pemrograman yang digunakan adalah visual studio code.

2.7.1 XAMPP

XAMPP adalah aplikasi web server yang berdiri sendiri terdiri dari Apache HTTP Server, MySQL *database* dan PHP. XAMPP dilengkapi dengan manajemen *database* PHPMyAdmin [24]. Untuk pembangunan sumber daya jenis kata, penulis menggunakan XAMPP sebagai web server. XAMPP memiliki kelebihan untuk bisa berperan sebagai server web Apache dalam melakukan simulasi pengembangan web. Tool pengembangan web berupa PHP, MySQL dan Perl. Dalam penelitian ini XAMPP digunakan sebagai server localhost untuk membuka sistem deteksi dan koreksi kesalahan ejaan *real-word* secara *offline*.

2.7.2 Web *Browser*

Web *Browser* adalah layanan internet untuk menjelajahi dunia maya dengan menggunakan jaringan internet [25]. Fungsi dari Web *Browser* sendiri adalah untuk menampilkan halaman web atau melakukan interaksi dengan dokumen yang disediakan server. Dalam penelitian ini penulis menggunakan Google Chrome versi 107.0.5304.107 sebagai w*eb browser* untuk pengembangan dan pengujian sistem deteksi dan koreksi kesalahan ejaan *real-word* yang dikembangkan.

2.7.3 Visual Studio Code

Visual Studio Code adalah program aplikasi yang berguna untuk mengedit teks dan skrip kode pemrograman seperti HTML, CSS, PHP, XML, Java, dan lain-lain. Visual Studio Code dapat berjalan pada sistem operasi Windows, Linux maupun macOS. Kelebihan visual studio code jika dibandingkan dengan sublime text yaitu gratis. Dalam penelitian ini visual studio code digunakan untuk mengedit kode program karena juga memiliki tampilan yang menarik dan juga memiliki fitur yang lengkap.