PENDETEKSIAN KESALAHAN KETIK DENGAN DAMERAU-LEVENSHTEIN DISTANCE DAN TRIE

James Tirta Halim [1] Lely Hiryanto [2] Irvan Lewenusa [3]

[1] Teknik Informatika, FTI, Universitas Tarumanagara
Jl. Letjen S. Parman no. 1, Jakarta 11440 Indonesia
james.535210025@stu.untar.ac.id [1] lelyh@fti.untar.ac.id [2] irvanl@fti.untar.ac.id [3]

ABSTRACT

Typographical errors are commonly found in text. Many applications implement a spell checking feature to detect and correct typographical errors. Spell checking requires an algorithm to calculate the similarity of two strings. This study compares Damerau-Levenshtein Distance and Trie in checking and correcting typographical errors in the names of function calls in source code based on the processing time and accuracy of the spelling correction. Accuracy is calculated by classifying the results of the spelling correction in a Confusion Matrix. This study shows that Trie is faster than Damerau-Levenshtein Distance, in which Trie's processing time took 10.07% of Damerau-Levenshtein Distance's. However, Damerau-Levenshtein Distance can correct more types of typographical errors than Trie, yielding an accuracy of 89.7% compared to 45.71%.

Key words

Damerau-Levenshtein Distance, Trie, Typographical error

1. Pendahuluan

Kesalahan ketik merupakan kesalahan pengejaan kata yang dibuat dalam proses pengetikan. Contohnya adalah pengejaan fungsi *string* standar C "mempcy" yang seharusnya dieja "memcpy". Kesalahan ketik merupakan sebuah hal yang sangat umum ditemukan, sehingga banyak perangkat lunak mempunyai fitur *spell checker*.

Spell checker akan membaca teks dan menguraikan kata-kata yang akan dicek dan mengecek setiap kata dengan daftar kata dengan pengejaan yang benar. Dalam ini, kata yang akan dicek adalah nama fungsi yang dipanggil, dan daftar katanya adalah daftar fungsi yang terdeklarasi. Jika terdapat sebuah fungsi yang tidak termasuk dalam daftar tersebut, maka fungsi tersebut dapat dinyatakan salah ketik. Selain itu, spell checker dapat menyarankan fungsi dengan pengejaan yang benar untuk fungsi yang salah ketik. *Spell checker* dapat menggunakan algoritma *approximate string matching* untuk menentukan kemiripan antara fungsi yang salah ketik dan fungsi yang terdapat dalam daftar tersebut, di mana fungsi-fungsi dengan kemiripan yang signifikan dapat digunakan sebagai fungsi yang akan disarankan *spell checker*.

Algoritma *approximate string matching* yang dibutuhkan sebuah *spell checker* adalah sebuah algoritma yang dapat menghitung kemiripan antara dua *string*.

2. Landasan Teori

2.1 Damerau-Levenshtein Distance

Damerau-Levenshtein Distance (DLD) digunakan untuk menghitung kemiripan antara dua *string*. DLD menghasilkan sebuah nilai, *edit distance*, yang merupakan jumlah operasi substitusi yang dibutuhkan untuk mentransformasi suatu *string* menjadi *string* lain. Semakin kecil nilai DLD, semakin mirip kedua *string*. Jika dua *string* sama persis, nilai DLDnya adalah nol. Operasi yang diperbolehkan oleh Damerau-Levenshtein Distance adalah *insertion*, *deletion*, *substitution*, dan *transposition*.

Insertion merupakan operasi penambahan karakter pada *string*. *Deletion* merupakan operasi penghapusan karakter pada *string*. *Substitution* merupakan operasi penukaran karakter dengan karakter lain pada *string*. *Transposition* merupakan operasi pembalikan dua karakter yang bersebelahan pada *string*. Berikut adalah rumus untuk Damerau-Levenshtein Distance. [3]

$$d_{a,b}(i,j) = \min \begin{cases} 0 & \text{if } i = j = 0, \\ d_{a,b}(i-1,j) + 1 & \text{if } i > 0, \\ d_{a,b}(i,j-1) + 1 & \text{if } j > 0, \\ d_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} & \text{if } i,j > 0, \\ d_{a,b}(i-2,j-2) + 1_{(a_i \neq b_j)} & \text{if } i,j > 1 \text{ and } a_i = b_{j-1} \text{ and } a_{i-1} = b_j, \end{cases} \quad (1)$$

Keterangan:

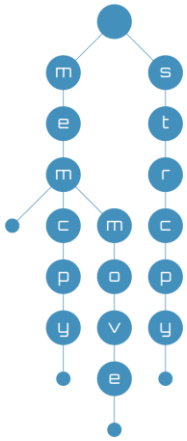
a, b = *string*
i, j = indeks karakter dalam string
a_i, b_j = $a[i]$, $b[j]$ = karakter ke-i dari suatu string a
$a_i \neq b_j$ = jumlah operasi substitusi; satu jika benar, nol jika sebaliknya

Untuk mengoreksi sebuah kesalahan ketik dengan DLD, bandingkan nilai DLD dari nama fungsi yang dicek dengan setiap nama fungsi yang dideklarasi. Jika ditemukan nilai DLD nol, maka tidak dideteksi kesalahan ketik. Nama fungsi yang dideklarasi dengan nilai DLD terkecil merupakan nama fungsi yang akan disarankan

sebagai pengoreksian. Dalam penelitian ini, ditetapkan batas nilai DLD dua, di mana nilai DLD lebih besar dari dua dinyatakan sebagai kesalahan ketik yang tidak dapat dikoreksi karena kemiripan dua *string* yang dibandingkan tidak terlalu signifikan.

2.2 Trie

Trie merupakan sebuah data struktur *tree* yang digunakan untuk mengecek jika suatu *string* terdapat pada sebuah daftar *string*. Trie juga dapat mengecek jika suatu *string* merupakan sebuah prefiks dari sebuah *string* dalam daftar *string*. Setiap *node* dalam Trie menyimpan sebuah karakter dari *string*. Dua atau lebih *string* dengan prefiks yang sama akan menggunakan *node-node* yang sama. Contoh struktur Trie yang berisi *string* "mem", "memcpy", "memmove", dan "strcpy" dapat dilihat pada Gambar 1. [2]



Gambar 1. Struktur Trie

Trie mendukung operasi *insertion*, *deletion*, dan *search*. Untuk mengoreksi sebuah kesalahan ketik dengan Trie, semua nama fungsi yang dideklarasi di-*insert* ke Trie. Nama fungsi yang akan dicek akan di-*search* pada Trie. Jika ditemukan *exact match*, maka tidak dideteksi kesalahan ketik. Jika ditemukan *prefix match*, maka dideteksi kesalahan ketik dan kunjungi sampai *node leaf* untuk memperoleh pengoreksiannya. Selainnya, dideteksi kesalahan ketik yang tidak bisa dikoreksi.

2.3 Confusion Matrix

Confusion Matrix merupakan sebuah metode klasifikasi performa algoritma. Performa tersebut diukur dengan membandingkan hasil yang diprediksi dari sebuah algoritma dengan hasil aktualnya. Terdapat empat jenis hasil, True Positive (TP), True Negative (TP), False Positive (FP), False Negative (FN). TP merupakan sebuah hasil positif yang diprediksikan positif. TN merupakan hasil negatif yang diprediksikan negatif. FP merupakan hasil negatif yang diprediksikan positif. FN merupakan hasil positif yang diprediksikan negatif. True Positive dan True Negative merupakan hasil yang sesuai prediksi dan False Positive dan False Negative merupakan hasil yang tidak sesuai hasil aktualnya. Confusion Matrix ini akan dipakai untuk mengevaluasi akurasi pengoreksian kesalahan ketik. Struktur dari Confusion Matrix dapat dilihat pada Tabel 1. [1]

Tabel 1 Confusion Matrix

		Kondisi Terprediksi	
	Total *population* = P + N	Positive (PP)	Negative (PN)
Kondisi Aktual	Positive (P)	True Positive (TP)	True Negative (TN)
	Negative (N)	False Positive (FP)	False Negative (FN)

2.4 Akurasi

Akurasi didapatkan dengan hasil-hasil yang diperoleh dari Confusion Matrix, True Positive (TP), True Negative (TN), Positive (P), dan Negative (N). Rumus dari akurasi adalah sebagai berikut.

$$ACC = \frac{TP + TN}{P + N} \quad (2)$$

Keterangan:

ACC = akurasi
TP = True Positive
TN = True Negative
P = Positive
N = Negative

3. Hasil Percobaan

Percobaan ini dilakukan dengan menggunakan tiga puluh sampel kode sumber dengan kesalahan ketik. Kesalahan ketik yang akan dideteksi adalah kesalahan dalam pengejaan nama pemanggilan fungsi. Kesalahan ketik ditentukan pada pemanggilan fungsi dengan nama fungsi yang belum dideklarasi. Hal yang akan diukur adalah kecepatan dan akurasi. Kecepatan diukur dengan membandingkan kecepatan eksekusi program metode dengan kecepatan eksekusi program metode DLD sebagai *baseline*. Dalam Confusion Matrix, klasifikasi kondisi TP merupakan pendeteksian kesalahan ketik yang memang salah ketik dengan pengoreksian yang benar, TN merupakan pendeteksian tidak salah ketik yang memang tidak salah ketik, FP merupakan pendeteksian salah ketik yang seharusnya tidak salah ketik dengan pengoreksian,

dan FN merupakan pendeteksian tidak salah ketik yang seharusnya salah ketik. Hasil Confusion Matrix dapat dilihat pada Tabel 2 dan Tabel 3.

Tabel 2 Confusion Matrix Damerau-Levenshtein Distance

Algoritma	Positif Terprediksi	Negatif Terprediksi
Positif Aktual	63	14
Negatif Aktual	0	59

Tabel 3 Confusion Matrix Trie

Algoritma	Positif Terprediksi	Negatif Terprediksi
Positif Aktual	4	76
Negatif Aktual	0	60

Perbandingan antara Damerau-Levenshtein Distance dan Trie dapat dilihat pada Tabel 4. Kecepatan yang disajikan adalah perbandingan antara kecepatan metode yang diuji dengan kecepatan DLD.

Tabel 4 Perbandingan Damerau-Levenshtein Distance dan Trie

Algoritma	Kecepatan	Akurasi
DLD	1	89,7 %
Trie	0.1007	45,71 %

4. Kesimpulan

Dari percobaan yang dilakukan, disimpulkan bahwa:

- Trie dan Damerau-Levenshtein Distance dapat mengoreksi kesalahan ketik.
- Trie dapat mengoreksi kesalahan ketik dengan jauh lebih cepat dibandingkan dengan Damerau-Levenshtein Distance.
- Damerau-Levenshtein Distance dapat mengoreksi lebih banyak jenis kesalahan ketik.

REFERENSI

[1] D. Chicco and G. Jurman, 'The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation', BMC genomics, vol. 21, pp. 1–13, 2020.

[2] Kvzaytsev, 'Title', GitHub, https://kvzaytsev.github.io/trie-visualizer/.

[3] L. Boytsov, 'Indexing methods for approximate dictionary searching: Comparative analysis', Journal of Experimental Algorithmics, vol. 16, p. 1, 2011.

James Tirta Halim seorang mahasiswa pada program studi Fakultas Teknologi Informasi di Universitas Tarumanagara.

Lely Hiryanto memperoleh gelar S.T. dari Universitas Tarumanagara tahun 2001, M.Sc. dari Curtin University of Technology tahun 2006, dan Ph.D. dari Curtin University tahun 2022.

Irvan Lewenusa memperoleh gelar S.Kom dari Institut Pertanian Bogor tahun 2008 dan M.Kom dari Universitas Budi Luhur tahun 2017. Saat ini sebagai dosen tetap Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara, Jakarta.