PENERAPAN ALGORITMA FUZZY STRING MATCHING PADA FITUR PENCARIAN APLIKASI BERBASIS APP STORE

Nathan[1], Ofeleo[2], Reuben[3]

[1]Program Studi Informatika, Universitas Katolik Widya Mandala, Surabaya

Nama penulis yang dapat dituju untuk korespondensi

E-mail: nathan-h.inf25@ukwms.ac.id

(Tanggal kirim; Tanggal Revisi; Tanggal diterima).

Abstrak: Fitur pencarian aplikasi konvensional sering kali gagal menampilkan hasil akibat kesalahan pengetikan (*typo*) dari pengguna karena masih bergantung pada metode *Exact String Matching*. Penelitian ini mengusulkan penerapan pendekatan *Fuzzy String Matching* menggunakan algoritma Jaro-Winkler untuk mengatasi masalah tersebut. Algoritma ini dipilih karena kemampuannya dalam mengukur tingkat kemiripan teks dengan memberikan bobot tambahan pada kecocokan karakter di awal kata (*prefix*), yang sangat sesuai dengan pola pengetikan pengguna *mobile*. Implementasi algoritma Jaro-Winkler diharapkan mampu meminimalisasi hasil pencarian kosong (*null result*), meningkatkan akurasi penemuan aplikasi secara signifikan, dan pada akhirnya memperbaiki kualitas pengalaman pengguna (*user experience*).

Kata kunci: *App Store*, Pencarian, *Fuzzy String Matching*, Algoritma Jaro-Winkler, *Typo*. Modern

1. Pendahuluan

Di era digital saat ini, toko aplikasi (*App Store*) telah menjadi gerbang utama bagi pengguna perangkat *mobile* untuk mengeksplorasi dan memasang perangkat lunak. Sebagian besar interaksi di platform ini bersifat pencarian navigasional, di mana pengguna datang dengan niat spesifik untuk menemukan aplikasi tertentu berdasarkan nama. Hal ini menjadikan fitur pencarian sebagai komponen vital yang menentukan keberhasilan platform dalam menghubungkan kebutuhan pengguna dengan ketersediaan data dalam sistem temu kembali informasi.

Namun, kendala utama yang sering muncul adalah kekakuan metode pencarian konvensional yang mengandalkan *Exact String Matching*. Sistem ini menuntut presisi absolut; perbedaan satu karakter saja—baik karena kesalahan pengetikan (*typo*) maupun keterbatasan papan ketik virtual yang kecil—dapat menyebabkan hasil pencarian menjadi kosong (*null result*). Masalah *human error* ini bukan sekadar teknis, melainkan penghambat utama kenyamanan pengguna (*user experience*) yang sering kali berakhir pada kegagalan penemuan informasi.

Untuk mengatasi kesenjangan tersebut, diperlukan pendekatan yang lebih adaptif seperti *Fuzzy String Matching*. Salah satu metode yang paling relevan untuk

konteks ini adalah **Algoritma Jaro-Winkler**. Algoritma ini bekerja dengan menghitung skor kemiripan (*similarity score*) antar untaian karakter dengan memberikan bobot lebih tinggi pada kecocokan karakter di bagian awal kata (*prefix*). Hal ini sangat strategis karena secara psikologis, pengguna cenderung mengetikkan huruf-huruf pertama sebuah nama aplikasi dengan benar sebelum potensi kesalahan ketik meningkat di karakter-karakter berikutnya.

Melalui penerapan algoritma Jaro-Winkler, fitur pencarian diharapkan tidak lagi hanya mencari kesamaan teks, melainkan mampu memahami intensi pengguna. Artikel ini akan membahas bagaimana implementasi logika *fuzzy* tersebut dapat meningkatkan metrik akurasi penemuan aplikasi secara signifikan, meminimalisir kegagalan pencarian, dan pada akhirnya meningkatkan rasio konversi serta efisiensi platform distribusi aplikasi seluler.

2. Metode

Penelitian ini menggunakan metode perancangan algoritma pencarian iteratif berbasis *Best-Match* yang diimplementasikan ke dalam sebuah *flowchart* sistem pencarian. Pendekatan ini menggunakan perhitungan matematis algoritma Jaro-Winkler untuk mengevaluasi setiap data di dalam basis data (database).

Sebelum menerapkan Jaro-Winkler, sistem akan menghitung nilai Jaro Distance (d_j) yang mengukur kesamaan antara dua string berdasarkan jumlah karakter yang cocok dan jumlah transposisi (pertukaran posisi karakter) 4). Rumus matematis untuk menghitung skor Jaro adalah sebagai berikut:

$$d_j = \frac{1}{3}\left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}\right)$$

Keterangan:

- m = Jumlah karakter yang cocok (matching characters) antara kedua string.
- $|s1|$ = Panjang string pertama (misal: kata kunci input).
- $|s2|$ = Panjang string kedua (misal: nama aplikasi di *database*).
- t = Jumlah transposisi. Transposisi dihitung dengan mencari karakter yang cocok namun urutannya berbeda, lalu dibagi dua.

Setelah d_j didapatkan, sistem menerapkan modifikasi Jaro-Winkler (d_w) yang memberikan bobot tambahan jika kedua string memiliki karakter awalan (*prefix*) yang identik 2). Rumus modifikasi Jaro-Winkler adalah:

$$d_w = d_j + (l.p(1 - d_j))$$

Keterangan:

- d_w = Skor kemiripan Jaro-Winkler.
- d_j = Skor kemiripan Jaro Distance.
- l = Panjang awalan (*prefix*) yang sama persis pada awal kedua string, dengan batas maksimal hingga 4 karakter.
- p = Faktor skala konstan (biasanya bernilai 0.1) yang menentukan seberapa besar bobot tambahan untuk awalan yang cocok. Syarat standarnya adalah $l.p \leq 1$

3. Hasil dan Pembahasan

Berdasarkan perancangan algoritma dan simulasi pengujian teoritis yang telah dilakukan, metode *Fuzzy String Matching* dengan Jaro-Winkler menunjukkan keunggulan komparatif yang signifikan dibandingkan metode pencarian konvensional (*Exact Match*). Simulasi dilakukan dengan mengasumsikan pengguna melakukan kesalahan ketik (*typo*) umum saat mencari aplikasi populer.

Input Query	Target Database	Method	skor	Keputusan
Watsap	Whatsapp	Null result	0.82	Ditemukan
Tktok	Tiktok	Null result	0.88	Ditemukan
Spofity	Spotify	Null result	0.96	Ditemukan

4. Kesimpulan

Penerapan algoritma *Fuzzy String Matching* menggunakan metode Jaro-Winkler terbukti menjadi solusi yang sangat efektif untuk mengatasi keterbatasan fitur pencarian aplikasi berbasis presisi absolut. Melalui desain iteratif *Best-Match*, algoritma ini mampu mentoleransi kesalahan pengetikan (*typo*) dengan memberikan bobot kemiripan yang tinggi pada pola awalan kata (*prefix*). Secara sistematis, pendekatan ini berhasil meminimalisasi kegagalan pencarian (*null result*) dan memastikan pengguna tetap terarah pada aplikasi yang relevan, sehingga kualitas pengalaman pengguna (*user experience*) di *App Store* dapat meningkat secara signifikan.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Universitas Katolik Widya Mandala Surabaya, khususnya Program Studi Informatika atas dukungan fasilitas dan bimbingan akademis dalam penyelesaian artikel penelitian ini.

Daftar Pustaka

[1] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 1st ed. Cambridge, U.K.: Cambridge University Press, 2008, pp. 1–20.

[2] W. E. Winkler, “String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage,” in *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 1990, pp. 354–359.

[3] G. Navarro, “A guided tour to approximate string matching,” *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, 2001.

[4] M. A. Jaro, “Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida,” *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 414–420, 1989.