

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era digital saat ini, sebuah perpustakaan sering kali menyediakan sebuah sistem untuk pencarian judul buku yang dapat memudahkan pengunjung untuk mencari buku yang di inginkan, namun dari hal ini muncul masalah umum yang sering dihadapi oleh penggunanya, yaitu sebuah kesalahan input kata kunci atau biasa disebut *typo (Typographical Error)*, Kesalahan pengetikan atau *typo* dapat disebabkan oleh berbagai faktor, seperti ketidaksesuaian antara media pengetikan dan fisik pengguna, adanya kerusakan pada perangkat pengetikan seperti keyboard maupun *touchscreen*, keterbatasan kosa kata pengguna, dan banyak faktor lainnya.

Fitur pencarian ialah salah satu fitur yang penting didalam sebuah aplikasi, karena adanya fitur pencarian dapat memudahkan pengguna dalam mencari sesuatu yang dicari tanpa perlu menelusuri setiap halaman yang ada. Cara kerjanya dengan menerima sebuah kata kunci yang di masukkan oleh pengguna, lalu *keyword* atau kata kunci tersebut diproses serta dicocokkan dengan sebuah data di *database*, jika data inputan terdapat di database, sistem akan menampilkan ke pengguna, namun jika sebaliknya, maka sistem akan menampilkan hasil kosong. Kata kunci ini dapat berupa kata atau frasa (beberapa kata)[1].

Sistem pencarian objek dengan tipe string telah mengalami banyak perkembangan, namun tidak jarang fitur-fitur pencarian ini hanya menampilkan hasil yang tepat sesuai dengan kata kunci yang dimasukkan. Hal ini berarti pengguna harus benar-benar memasukkan kata kunci yang sudah pasti benar, karena satu kesalahan pengetikan saja dapat menghasilkan output yang tidak sesuai atau bahkan tidak menampilkan hasil sama sekali. Saat ini, sedang dikembangkan sistem pencarian yang mampu memperkirakan kesalahan yang mungkin terjadi saat pengguna memasukkan kata kunci, sehingga dapat menangani masalah jika pengguna memberikan kata kunci yang salah atau tidak tepat. Jadi sebuah sistem yang dimaksud dapat memperbaiki atau mengoreksi kesalahan kata kunci yang

dimasukkan pengguna dengan membandingkan string-string yang terlibat dalam pencarian. Pembandingan ini dibagi menjadi dua, yaitu *exact string matching* dan *inexact string matching* [2].

Untuk membuat sebuah koreksi objek string, dapat menggunakan sebuah Algoritma *Levenshtein distance*. Algoritma ini merupakan sebuah algoritma yang digunakan untuk mencari sebuah jarak *Levenshtein* antara dua buah objek string yang diketikkan oleh pengguna (*string sumber*) sebuah string yang ada pada database (*string target*) sehingga mampu menampilkan sebuah koreksi kata untuk pencarian yang memiliki distance terdekat dengan string yang dicari dalam sebuah sistem pencarian[3]. Algoritma *Levenshtein distance* beroperasi dengan melakukan perhitungan jumlah jarak antar karakter yang berbeda antara string sumber dan string target. Hasil dari perhitungan tersebut diperoleh dengan cara menentukan jumlah minimum operasi yang diperlukan untuk mengubah suatu string menjadi string lainnya. Terdapat tiga macam operasi yang dapat dilakukan oleh algoritma *Levenshtein distance*, salah satunya adalah operasi penghapusan karakter atau *Deletion*, penggantian karakter atau *Substitution*, serta penambahan karakter atau *Insertion*[4].

Adapun terdapat penelitian - penelitian untuk proses deteksi kata yang salah dalam pengejaan dan memberikan saran kata yang benar. Penelitian tersebut antara lain Aplikasi Pengarsipan Dokumen Menggunakan Metode *String Matching*, dimana penelitian ini bertujuan untuk memberikan saran-saran untuk perbaikan salah ketik di fitur pencarian, dimana koreksi pencarian kata kunci diperoleh dengan menggunakan Algoritma *Damerau-Levenshtein distance*[1], selanjutnya ada penelitian tentang metode *Levenshtein distance* dan *N-Gram* Berdasarkan hasil pengujian dan analisis identifikasi *typo* pada dokumen berbahasa Indonesia yang menggunakan metode *N-Gram* dan *Levenshtein distance*, penelitian selanjutnya menunjukkan bahwa pendekatan Dictionary Lookup dalam proses identifikasi *typo* pada dokumen bahasa Indonesia dapat diterapkan dengan baik.[5].

Berdasarkan uraian yang telah dijelaskan diatas, maka Penelitian ini bertujuan untuk membangun sebuah sistem pencarian dengan fitur koreksi kata yang

menggunakan algoritma *Levenshtein distance*, dimana diharapkan nantinya dengan adanya fitur ini dapat mengoreksi kata serta membantu memberikan sebuah saran kata dari kata terdekat yang di ajukan atau dimasukkan oleh pengguna, sehingga jika ada pengguna yang melakukan kesalahan dalam memasukkan kata kunci, maka sistem masih bisa menampilkan kata terdekat dari kata yang di inputkan oleh pengguna itu sendiri.

1.2 Perumusan Masalah

Terdapat tiga hal didalam perumusan masalah, yaitu permasalahan, metode usulan, serta pertanyaan penelitian.

1.2.1 Permasalahan

Berdasarkan latar belakang diatas, maka muncul sebuah permasalahan yaitu sebuah sistem pencarian objek dengan tipe string yang kini telah banyak dikembangkan dan hanya dapat memberikan hasil pencarian yang sesuai dengan kata kunci yang dimasukkan, seperti pada sistem pencarian buku yang ada di perpustakaan Universitas Trunojoyo Madura, padahal kesalahan dalam pengetikan kata kunci atau biasa disebut *typo* (*Typographical Error*) ini bisa saja terjadi, karena disebabkan oleh beberapa hal, seperti media pengetikan yang tidak sesuai dengan fisik pengguna, kerusakan pada media pengetikan, minimnya kosakata pengguna dan lain sebagainya

1.2.2 Metode Usulan

Metode yang di usulkan disini ialah menggunakan Algoritma *Levenshtein distance*, dimana algortima ini digunakan untuk mencari sebuah jarak antara dua buah objek string yang diketikkan oleh pengguna (*string* sumber) dengan sebuah string yang ada pada database (*string* target) sehingga mampu menampilkan sebuah koreksi kata untuk pencarian yang memiliki distance terdekat dengan kata kunci yang dicari dalam sebuah sistem pencarian.

1.2.3 Pertanyaan Penelitian

Berdasarkan dengan latar belakang serta rumusan masalah diatas maka menghasilkan pertanyaan penelitian yaitu seberapa besar nilai *precision* dan *recall* dalam penggunaan algoritma *Levenshtein distance* untuk mengoreksi sebuah kesalahan pengetikan kata kunci didalam sebuah fitur pencarian?

1.3 Tujuan dan Manfaat Penelitian

1.3.1 Tujuan Penelitian

Tujuan dari penelitian ini untuk mengetahui seberapa besar *precision* dan *recall* dalam penggunaan metode algoritma *Levenshtein distance* untuk mengoreksi sebuah kesalahan pengetikan *keyword* didalam sebuah sistem pencarian.

1.3.2 Manfaat Penelitian

Penelitian ini diharapkan mampu untuk :

1. Membantu mengoreksi kata kunci yang di inputkan oleh pengguna melalui saran kata yang dihasilkan, sehingga apabila terdapat kesalahan pengetikan, sistem masih dapat menampilkan sebuah hasil data melalui kata kunci terdekat dari inputan yang dilakukan oleh pengguna
2. Sebagai bahan pembelajaran dan refrensi untuk peneliti selanjutnya.

1.4 Batasan Masalah

Berikut merupakan batasan masalah dari penelitian ini :

- a Data berjumlah 240 judul buku berdasarkan kategori *computer program* dan *software* yang ada di perpustakaan Universitas Trunojoyo Madura.
- b Batas judul buku yang relevan berjumlah 10 judul buku
- c Penggunaan algoritma *Levenshtein distance* dalam pencarian disini dibatasi hanya menggunakan *threshold* atau ambang batas ≤ 2 huruf saja.
- d Saran kata yang muncul tidak memperhatikan konteks atau makna dari kata kunci
- e *Output* dari sistem ini nanti hanya berupa sistem pencarian.

f Data hanya meliputi judul buku saja.

1.5 Metodologi Penulisan Skripsi

Dalam menyusun laporan ini, digunakan sistematika penulisan yang telah disesuaikan berdasarkan format penulisan Skripsi pada Program Studi Sistem Informasi Fakultas Teknik Universitas Trunojoyo Madura:

BAB I PENDAHULUAN

Pada bab ini memaparkan tentang latar belakang, perumusan masalah, tujuan dan manfaat penelitian, Batasan masalah, dan sistematika penulisan

BAB II TINJAUAN PUSTAKA

Pada bab ini memaparkan tentang tinjauan Pustaka, teori-teori yang digunakan yang terkait dengan tema penelitian

BAB III METODE PENELITIAN

Pada bab ini memaparkan mengenai metode yang digunakan, yaitu metode algoritma *Levenshtein distance*.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini memaparkan mengenai implementasi serta pengujian pada sistem yang telah dibuat yang nantinya akan ditarik kesimpulan pada bab V.

BAB V PENUTUP

Pada bab ini terdiri dari kesimpulan dan saran, kesimpulan memberikan penjelasan mengenai hasil dari penelitian terhadap rumusan masalah. Saran berisi rekomendasi untuk peneliti selanjutnya