

Penerapan Algoritma String Matching dan Regular Expression pada Aplikasi Kamus Besar Bahasa Indonesia (KBBI)

Jasmine Mutiara Bintang¹, Muhammad Faisal Ashshidiq², Hilal Fakhri Dzakhwan³

¹Teknik Informatika, Universitas Logistik dan Bisnis Internasional, 1214012@std.ulbi.ac.id

²Teknik Informatika, Universitas Logistik dan Bisnis Internasional, 1214041@std.ulbi.ac.id

³Teknik Informatika, Universitas Logistik dan Bisnis Internasional, 1214018@std.ulbi.ac.id

Keywords:

*Big Indonesian Dictionary (KBBI),
String Matching Algorithm,
Regular Expression Algorithm*

ABSTRACT

The Kamus Besar Bahasa Indonesia (KBBI) application is a software or application that allows users to access and search for the meaning of a single word in Indonesian along with its spelling. Indonesian has a complex morphological structure, where words can change form through affixation (addition of prefixes or suffixes), reduplication, or internal changes that create many dialectal variations and spelling variations that can affect word forms and writing patterns. This causes a variety of words that are similar but different in writing. On the other hand, that the Indonesian Dictionary application needs speed and accuracy of word searches then the program makes adjustments to those in the dictionary. Therefore, researchers analyze based on the data that has been obtained by the KBBI application using string matching (brute force) and regular expression algorithms. Researchers hope that the algorithms that have been analyzed can help solve problems in the KBBI application in conducting word searches to find the meaning of spelling funds.

Kata Kunci

*Kamus Besar Bahasa Indonesia (KBBI),
Algoritma String Matching,
Algoritma Regular Expression,*

ABSTRAK

Aplikasi Kamus Besar Bahasa Indonesia (KBBI) adalah sebuah perangkat lunak atau aplikasi yang memungkinkan pengguna untuk mengakses dan mencari makna satu kata dalam bahasa Indonesia beserta ejaannya. Bahasa Indonesia memiliki struktur morfologis yang kompleks, di mana kata kata dapat mengalami perubahan bentuk melalui afiksasi (penambahan awalan atau akhiran), reduplikasi, atau perubahan internal yang membuat banyak variasi dialek dan variasi ejaan yang dapat mempengaruhi bentuk kata dan pola penulisan. Hal ini menyebabkan variasi kata yang mirip tetapi berbeda dalam penulisan. Dilain sisi, bahwa aplikasi Kamus Bahasa Indonesia perlu kecepatan dan akurasi pencari kata kemudia program melakukan penyesuaian dengan yang ada di kamus. Maka dari itu peneliti menganalisa berdasarkan data yang telah di dapat aplikasi KBBI menggunakan algoritma string matching (brute force) dan regular expression. Harapan peneliti algoritma yang telah di Analisa dapat membantu memecahkan masalah pada aplikasi KBBI dalam melakukan pencarian kata untuk mencari makna dana ejaan

Korespondensi Penulis:

Jasmine Mutiara Bintang,
Universitas Logistik dan Bisnis Internasional,
Gedung Rektorat, Sariasih No. 54, Kota Bandung
Telepon: +6282363219861
Email: 1214012@std.ulbi.ac.id

1. PENDAHULUAN

Semakin maraknya perkembangan zaman khususnya dibidang teknologi maka semakin banyak pula orang yang membangun atau merancang sebuah aplikasi, contohnya dalam dunia pendidikan kamus bahasa indonesia sangat dibutuhkan khususnya dalam mata pelajaran yang membutuhkan kamus bahasa Indonesia[1]. Aplikasi kamus besar bahasa indonesia adalah perangkat lunak yang sudah disediakan baik dari segi kolom pencarian atau *search* atau berdasarkan inputan kata yang dicari dan juga beserta arti arti makna dari kata yang dicari didalam kamus Bahasa indonesia tersebut[2]. Sebuah kamus digital adalah sejenis rekayasa perangkat lunak yang dibuat untuk menyelesaikan sebuah persoalan dalam pencarian kata dengan cepat dan efisien menggunakan mesin pencari dan juga berdasarkan urutan urutan mulai dari huruf A sampai Z yang memiliki makna tersendiri[3]. Kamus besar Bahasa Indonesia (KBBI)

saat ini juga semakin marak digunakan didunia Pendidikan baik jenjang sekolah maupun perkuliahan yang tetap berguna dikalangnya[4].

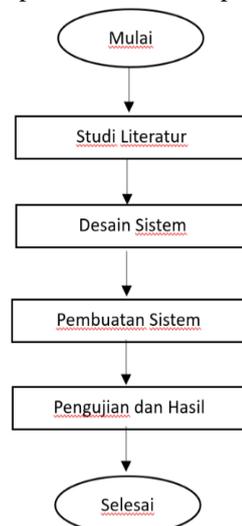
Algoritma pencocokan *String* atau kata (*String Matching*) adalah sebuah pencarian suatu kata kata dalam Bahasa Indonesia misalnya kata “khilaf” yang mempunyai makna berarti suatu tindakan yang dilakukan dengan tidak sengaja[5]. Pencocokan *string* tentunya sangat dibutuhkan dan digunakan dalam media patokan bagi seluruh pengguna dalam menambah pengetahuan terkait kata yang sering diucapkan sehari hari[6]. Namun, bahasa Indonesia memiliki beberapa karakteristik yang membuat analisis dan pencocokan kata menjadi tantangan[2]. Bahasa Indonesia memiliki struktur morfologis yang kompleks, di mana kata kata dapat mengalami perubahan bentuk melalui afiksasi (penambahan awalan atau akhiran), reduplikasi, atau perubahan internal yang membuat banyak variasi dialek dan variasi ejaan yang dapat mempengaruhi bentuk kata dan pola penulisan. Hal ini menyebabkan variasi kata yang sama atau mempunyai kemiripan tetapi berbeda dalam penulisan, misalnya "bercakap" dan "berbicara", atau "mengecat" dan "melukis". Dilain sisi, terdapat beberapa orang yang di era digital tetapi tetap menggunakan kamus buku Bahasa Indonesia yang saat ini digunakan untuk mencari kata per kata dengan membolak balikkan lembar buku tersebut sehingga terkesan menghabiskan banyak waktu.

Berdasarkan masalah yang dijelaskan diatas, tentunya juga sangat penting bagi setiap individu untuk memahami kata dalam bahasa indonesia baik kata dasar, kata awalan, akhiran maupun berimbuhan dengan memanfaatkan teknologi terkini yaitu aplikasi kamus besar bahasa indonesia[3]. Tujuan dari adanya suatu penelitian ini yakni menerapkan algoritma *String Matching* dan *Regular Expression* yang mampu mencocokkan kata dalam bahasa Indonesia serta dapat membantu dalam mencari kata dalam bahasa indonesia sehingga dapat meningkatkan kemampuan berbahasa Indonesia yang baik. Metode yang digunakan dalam penerapan aplikasi kamus besar Bahasa Indonesia (KBBI) tersebut adalah dengan algoritma *string matching* dan *regular expression*[2]. Maka dari itu, penelitian ini diharapkan mampu mengetahui penerapan algoritma *string matching* dan *regular expression* yang digunakan dalam aplikasi KBBI sehingga dapat dibandingkan antara keduanya.

2. METODE PENELITIAN

2.1 Metode Pengembangan Perangkat lunak

Metode penelitian adalah urutan yang biasanya dipakai dalam menyelesaikan suatu penelitian yang terdapat permasalahannya[7]. Tahapan dalam diagram alur penelitian ini terdapat beberapa diantaranya sebagai berikut:



Gambar 2.1. Diagram Alur Penelitian

Dari gambar diatas adalah sebuah tahapan dalam metode penelitian diantaranya adalah dijelaskan pada uraian berikut ini:

1) Studi Literatur

Studi kepustakaan berisikan mengenai *research* melalui berbagai studi literatur dari jurnal, buku atau referensi terakait penelitian yang digunakan serta hasil penelitian yang ada dalam jurnal penelitian sebelumnya[1].

2) Desain Sistem

Pada tahapan ini dilakukan serangkaian proses untuk melakukan proses desain sebuah aplikasi Kamus Besar Bahasa Indonesia (KBBI) dengan algoritma yaitu *string matching* dan *regular expression*[2].

3) Pembuatan Sistem (*Coding*)

Pada tahap ini dilakukan membuat sistem dengan mencantumkan kode program yang terlibat[2].

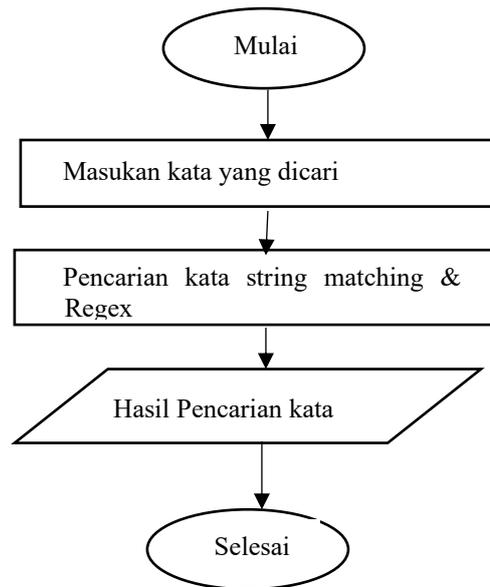
4) Pengujian

Pada tahap yang ke 4 ini adalah tahap dengan melakukan pengujian aplikasi KBBI guna untuk mendeteksi kelayakan aplikasi kamus besar Bahasa Indonesia serta melihat eksekusi program[3].

5) Hasil

Pada tahap ini hasil dari pengujian aplikasi yang dijalankan apakah dapat berjalan maksimal seperti yang diharapkan.

2.2 Flowchart Sistem



Gambar 2.2 Flowchart Gambaran Umum Sistem

Gambar 2.2 merupakan tahapan dari gambaran *flowchart* sistem secara umum, dimana terdapat sebuah tahapan yakni diawali dengan mulai program, kemudian masukan kata yang dicari misalkan “hujan” setelah mengetik kata “hujan” akan di proses dengan menggunakan algoritma string matching maka kamus akan menampilkan hasil kata dari hujan secara spesifik dan jelas dengan menggunakan metode regex dan selesai.

2.3 String Matching & Regex

Aplikasi Kamus Bahasa Indonesia adalah tools untuk mempermudah banyak orang dalam mencari sebuah kata yang belum diketahui maknanya, selain itu dapat mengetahui huruf ejaan yang berawal dan berakhir. Contoh kata berawal seperti be, me, ke dll. Sedangkan contoh kata berakhir seperti ng, an, in dll. Sesuai aturan Bahasa Indonesia yang baku ataupun tidak baku. Hal tersebut aplikasi menggunakan algoritma yang disebut string matching dan regex. String matching yaitu metode dalam pencocokan dalam suatu kata, kalimat atau string untuk pencarian kata kunci yang ingin di cari. Sedangkan regex (Regular Expression) yaitu suatu pola yang berurutan dari beberapa karakter yang digunakan dalam pencocokan kata kunci.

Algoritma string matching dan regex telah di analisis, maka data yang di dihasilkan berupa alur dari kedua algoritma tersebut, berikut beberapa alur yang dipahami dari aplikasi KBBI:

- String Matching: proses pencocokan kata dari mulai penginputan kata kunci dari pihak pengguna akan di terima oleh server. Setelah merespon oleh server akan proses memuat data yang ada di kamus. Kamus di sini yakni kumpulan kata atau string untuk menyesuaikan kata kunci yang ingin dicari, pengguna sudah melakukan pencarian kata kunci suatu kata maka program akan membandingkan kata kunci dengan yang ada di kamus setelah cocok kata kunci yang ingin dicari oleh pengguna maka akan menampilkan hasil.
- Regex: proses pencocokan pola yang ada di suatu kata kunci dari mulai penginputan kata kunci dari pihak pengguna akan di terima oleh server. Setelah di terima server akan melakukan pencocokan kata sesuai dengan kamus merespon sesuai data yang ada di kamus, setelah cocok kata kunci yang dicari oleh pengguna tahap selanjutnya akan membuat pola untuk mengetahui makna dari kata kunci yang di cari.

3 HASIL DAN ANALISIS

Setelah melakukan eksperimen dan membandingkan antara kedua algoritma tersebut, terdapat sebuah hasil yang didapatkan dari kode program yang dijalankan atau *dirunning*. Adapun rancangan sistem dalam aplikasi kamus besar Bahasa Indonesia dapat dijabarkan pada pembahasan dibawah ini:

3.1 Hasil Pengujian Pencarian *String* Pada Kamus Besar Bahasa Indonesia (KBBI) Dengan Algoritma *String Matching*

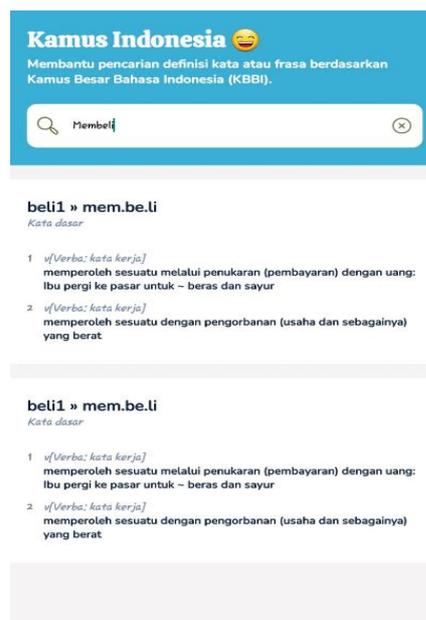
Pada aplikasi KBBI cara pengujiannya dilakukan dengan menerapkan algoritma yang pertama yakni *string matching* apabila pengguna atau pemakai aplikasi mengetikkan *keyword* yang dicari dikolom *search*, maka hasil analisis yang dapat dirangkum seperti dibawah ini:

- 1) Pada pengujian *string matching* yang diterapkan yaitu algoritma *brute force* dimana yang pada dasarnya menerapkan proses dari kiri ke kanan yang gunanya untuk mencocokkan *string* satu per satu kata.

- 2) Jika terdapat sebuah kata saja yang tidak masuk dalam kriteria pencocokkan berdasarkan kata yang dicari, maka sebuah iterator pada kata *string* yang dicocokkan akan diterapkan secara berulang yakni mulai kiri ke kanans pola sampai menemukan *matching* atau kecocokkan.
 - 3) Terakhir, aplikasi KBBI akan menampilkan sebuah *result* (hasil) dari proses iterator sebelumnya dan menunjukkan hasil kata sesuai dengan pencariannya.
- Contoh:



Gambar 3.1 Contoh *String Matching* Dengan Algoritma *Brute Force*

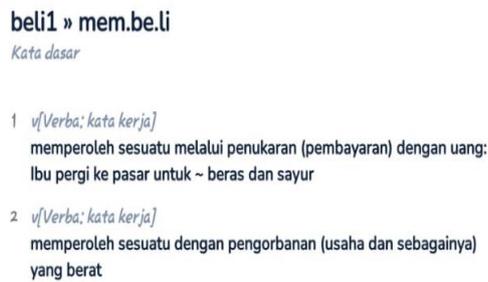


Gambar 3.2 Contoh *String Matching* Pada KBBI

Pada gambar 3.2 dijabarkan bahwa penerapan string matching pada Kamus Besar Bahasa Indonesia (KBBI) melibatkan pencarian kata kata dalam kamus berdasarkan kesamaan urutan karakter atau *string*. *String matching* dapat melakukan pencarian kata kata berdasarkan pola yang cocok dengan kata kunci yang diberikan. Contoh kata "membeli" seperti contoh diatas dapat digunakan untuk menjelaskan penerapan string matching pada KBBI. Dalam kasus ini, *string matching* akan mencari kata "membeli" dalam kamus KBBI untuk memverifikasi keberadaan kata tersebut dan mendapatkan informasi terkait definisi, ejaan, atau informasi lain yang terkait dengan kata tersebut. Berikut adalah penjelasan penerapan *string matching* pada KBBI dengan kata "membeli" seperti yang dicontohkan:

- a.) Pertama, kita mengambil kata "membeli" sebagai kata kunci yang ingin dicari dalam kamus KBBI. Kemudian, kita melakukan *string matching* dengan membandingkan kata kunci "membeli" dengan setiap kata dalam kamus KBBI. Proses *string matching* akan mencari kecocokan antara karakter dalam kata kunci dan setiap kata dalam kamus KBBI.
- b.) Jika ada kecocokan yang ditemukan, kata "membeli" dianggap cocok dengan kata dalam kamus KBBI. Setelah kecocokan ditemukan, kita dapat mengambil informasi terkait dengan kata "membeli" seperti definisi, sinonim, atau penggunaan kata yang benar. Misalnya, setelah melakukan *string matching* pada KBBI, kita menemukan kata "membeli" dalam kamus. Hasil pencarian akan memberikan informasi seperti:
 - Kata: membeli
 - Ejaan: mem.be.li
 - Kata dasar: beli
 - Kelas kata: verba (kata kerja)

- Arti: melakukan pembelian; mendapatkan sesuatu dengan menukar uang atau barang lain yang dianggap senilai.



Gambar 3.3 Contoh *String Matching* Kata Membeli

3.2 Hasil Pengujian Pencarian *String* Pada Kamus Besar Bahasa Indonesia (KBBI) Dengan Algoritma *Regular Expression*

Algoritma *Regular Expression* adalah sebuah algoritma yang menerapkan suatu *heuristic matching* yang dapat mencocokkan pola per kata yang dicari. Contoh kasusnya, apabila seorang pengguna memasukkan sebuah kombinasi antara dengan menggunakan *keyword* yang sama persis dan memasukkan metode *Regular Expression* maka hasilnya sebagai berikut:

- 1.) Aplikasi KBBI otomatis akan mencari sekaligus dengan menggunakan per kata dalam sebuah *string* yang dicari. contohnya "Pisau".
- 2.) Namun, apabila pola *pattern* didalam semua kata tersebut terdapat didalam sebuah teks, maka fungsi pencarian *regular expression* akan mengembalikan *true* atau benar. Jika tidak benar maka akan mengembalikan sebuah *false*.



Gambar 3.4 Contoh Kata Pisau *Regular Expression* Pada KBBI

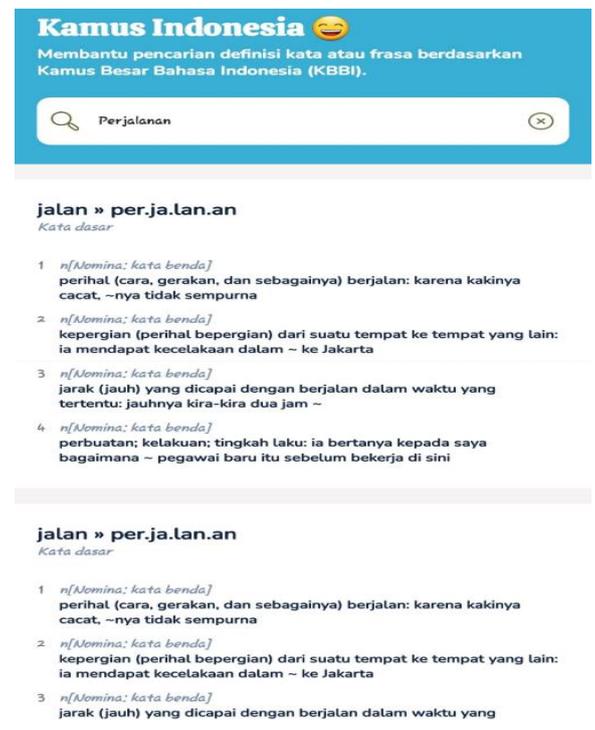
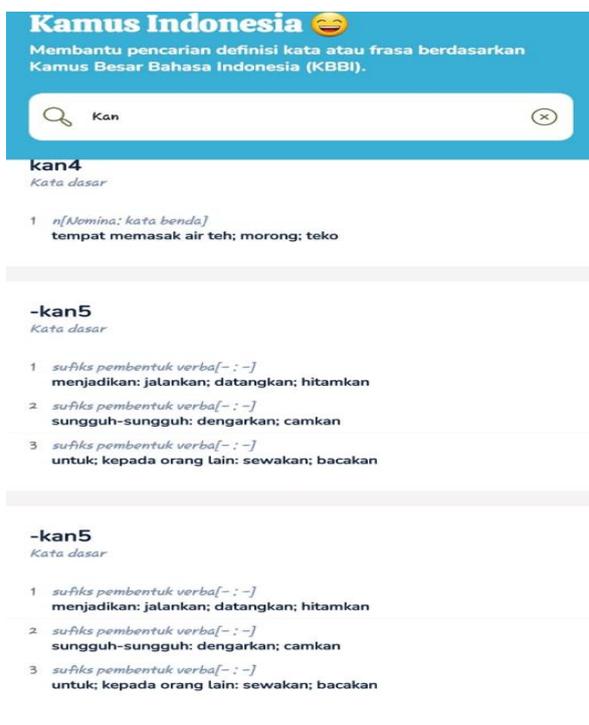
Penerapan algoritma *regular expression* pada Kamus Besar Bahasa Indonesia (KBBI) melibatkan penggunaan pola pola khusus untuk mencocokkan dan mencari kata-kata yang memenuhi pola yang didefinisikan. *Regular expression* (regex) adalah alat yang kuat untuk memanipulasi dan mencocokkan teks berdasarkan pola tertentu. Misalnya, contoh diatas menggunakan kata "pisau" adapun langkah penerapan algoritma *regular expression* pada KBBI dengan cara sebagai berikut:

- 1) Pertama mendefinisikan pola *regular expression* yang ingin dicocokkan dengan kata kata dalam KBBI. Dalam kasus ini, jika ingin mencocokkan kata kata yang memiliki pola "pisau". Pola *regular expression* yang umum digunakan adalah dengan menggunakan karakter khusus seperti huruf besar dan kecil, karakter spesial, dan metakarakter. Pola yang cocok dengan kata "pisau" adalah "pisau".
- 2) Kedua mendefinisikan pola, kita menerapkannya pada kamus KBBI menggunakan algoritma *regular expression*. Proses ini akan mencocokkan pola "pisau" dengan setiap kata dalam kamus KBBI. Jika ada kata dalam kamus yang cocok dengan pola "pisau", maka kata tersebut akan ditemukan dan bisa diambil informasinya.

Saat menerapkan algoritma *regular expression*, terdapat beberapa pola yang sering digunakan dalam KBBI. Berikut adalah beberapa contoh pola *regular expression* (regex) yang ada didalam Kamus Besar Bahasa Indonesia (KBBI) yakni seperti dibawah ini:

- 1) Pola: ".*kan\$"

- Contoh kata: membacakan, menghentikan, melompatkan
- 2) Pola: `^me.*kan$`
Contoh kata: memperkuat, mengecilkan, menyembuhkan
- 3) Pola: `^per.*an$`
Contoh kata: perjalanan, perubahan, persamaan
- 4) Pola: `.*i.i.`
Contoh kata: bintang, indikasi, lingkungan
- 5) Pola: `.*a.*i$`
Contoh kata: babakbelur, beterbangan, cemasai
- 6) Pola: `.*uI$`
Contoh kata: panjangul, terdalamul, tanggul
- 7) Pola: `^di.i.$`
Contoh kata: diundangi, dikunjungi, direnungi



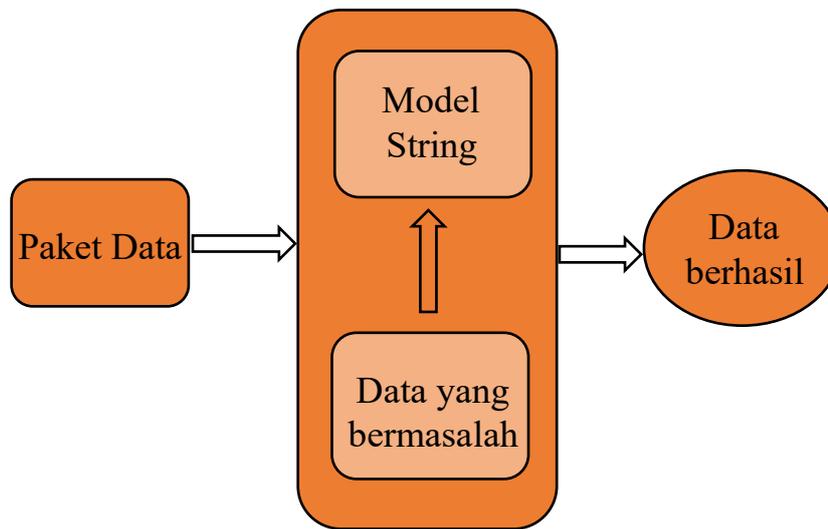
Gambar 3.5 Contoh Pola *Regular Expression* Pada KBBI

Dalam contoh di atas, pola "pe-an" berhasil mencocokkan kata "perjalanan" karena kata tersebut memiliki akhiran "per", diikuti oleh huruf "a", dan diakhiri dengan "an". Dengan menggunakan pola ini, kita dapat dengan cepat mengidentifikasi kata-kata dalam KBBI yang memiliki struktur yang sama dan membantu dalam mencari kata benda yang terkait dengan kata kerja tertentu. Sedangkan pola `.*kan$` mencocokkan kata kata dalam KBBI yang diakhiri dengan "kan". Misalnya, kata "menjadikan", "datangkan", dan "hitamkan" cocok dengan pola tersebut karena akhirnya adalah "kan".

3.3 Hasil Pengujian Kecepatan *String* Pada Kamus Besar Bahasa Indonesia (KBBI) Dengan Algoritma *String Matching dan regex*

a. String Matching (Brute Force)

Algoritma pencocokan string atau kata (String Matching) adalah salah satu algoritma yang sangat penting dalam pemrosesan teks dan digunakan untuk mencari kecocokan dalam sebuah string. Algoritma ini memiliki aplikasi luas dalam berbagai bidang, seperti pengolahan bahasa alami, analisis teks, pemrosesan log, dan masih banyak lagi. Dalam konteks bahasa Indonesia, algoritma pencocokan string sangat berguna dalam analisis teks dan pencarian kata-kata. Bahasa Indonesia memiliki karakteristik yang kompleks, termasuk variasi ejaan, variasi dialek, dan perubahan bentuk kata melalui afiksasi, duplikasi, atau perubahan internal. Pencocokan string memungkinkan pemrosesan yang akurat dan efisien untuk menemukan kata-kata yang relevan dalam teks bahasa Indonesia.



Gambar 3.6 Sistem Pendeteksi string

Metode string matching dengan brute force akan melakukan pencocokan beberapa karakter sesuai perintah oleh pengguna dari kiri sampai ke kanan inputan pencarian. Algoritma yang bisa digambarkan pada Gambar 3.6 yang mempunyai makna proses penginputan data yang dicari akan di sesuaikan dengan kamus dari kemendikbud ketika ada yang bermasalah maka akan disimpan ke model string sebagai bahan apabila kata kunci dicari kembali akan mengalami masalah karena tidak sesuai, Apabila berhasil maka akan menampilkan data-data yang sesuai dengan kamus hingga makna, penjelasan dari kata kunci yang dicari oleh pengguna.

b. Regex (Regular Expression)

Regex adalah sebuah notasi khusus yang digunakan untuk mencocokkan pola dalam teks. Dengan regex, kita dapat membuat pola yang kompleks untuk mencari kata-kata dengan karakteristik tertentu. Misalnya, kita dapat menggunakan regex untuk mencari semua kata yang diawali dengan huruf "a" dan diakhiri dengan huruf "t" dalam sebuah teks.

- ϵ adalah regular expression untuk bahasa reguler $\{\epsilon\}$.
- Apabila a ada simbol $\in \Sigma$ (Σ mewakili alfabet masukan dari inputan pencarian), maka a adalah regular expression dengan bahasa yang bisa dipahami $\{a\}$.
- Apabila a dan b adalah regular expression, maka $a + b$ juga merupakan ekspresi reguler dengan bahasa bisa dipahami $\{a,b\}$.
- Apabila a dan b adalah regular expression, maka ab (rangkaian a dan b) juga reguler.
- Apabila a adalah regular expression, a^* (0 kali atau lebih a) juga reguler.

	Regular Expression	Bahasa Regular
Kumpulan Vokal	$(a \cup e \cup i \cup o \cup u)$	$\{a,i,u,e,o\}$
A diikuti oleh b lebih	(ab^*)	$\{a,b, abb, abbb\dots\}$
Tidak ada.	$v^* .c^*$ (di mana v – vokal dan c – konsonan)	$\{ \epsilon , a ,aou, aiou, b, abcd\dots\}$ di mana ϵ mewakili string kosong (dalam kasus 0 vokal dan o konsonan)

Suatu bahasa yang beraturan apabila mempunyai aturan $A \rightarrow Z$ yang dimana ϵ adalah simbol khusus yang disebut NULL

c. Perbedaan string biasa dengan string matching & regex

Perbedaan antara pencarian biasa (query) dengan penerapan string matching dan regex yaitu, Pencarian Biasa (Query): Pencarian biasa dilakukan dengan menggunakan kata kunci atau frasa tertentu untuk mencari kecocokan dalam teks atau dokumen. String Matching: Algoritma pencocokan string memperhatikan variasi dalam penulisan kata dan pola penulisan yang mirip. Misalnya, jika kita mencari kata "khalif" menggunakan algoritma pencocokan string, kita akan mendapatkan hasil yang relevan meskipun kata tersebut ditulis dengan variasi seperti "khalif", "kilap", atau "klaf".

d. Kecepatan & Akurat

Setelah dilakukan analisis dari kedua algoritma yakni string matching secara kinerja cepat dan akurasi keduanya bergantung dari pemakaian atau berbagai faktor contohnya kasus dari pencarian pada kamus Bahasa Indonesia. Mulai dari ukuran karakter yang diketikan dalam elemen pencarian. String matching dan regex secara analisis bahwa keduanya saling bekerja sama untuk meningkatkan efisiensi pencarian makna dari satu kata. Dari segi akurasi algoritma string matching dan regex dapat akurat apabila digunakan secara benar. Penggunaan

secara benar bisa berupa proses pencocokan satu kata dengan lebih spesifik. Karena itu dapat di sarankan bahwa perlu mempertimbangkan persyaratan khusus pada kasus dari penggunaan metode yang akan digunakan untuk mencapai performa terbaik. Dengan menggunakan algoritma pencocokan string dan regex, kita dapat meningkatkan akurasi dan efisiensi dalam pencarian kata-kata atau pola tertentu dalam teks. Ini sangat membantu dalam analisis teks, pemrosesan bahasa alami, dan berbagai aplikasi lain yang melibatkan pemrosesan teks.

4 KESIMPULAN

Berdasarkan penelitian terkait penerapan algoritma *string matching* dan *regular expression* pada aplikasi kamus besar Bahasa Indonesia (KBBI), maka dapat ditarik sebuah kesimpulannya adalah sebagai berikut:

1. Pada algoritma *string matching* diperlukan sebuah *keyword related* misalkan “hujan” maka hasil pencarian yang dapat menemukan semua kata yang dicari dengan jumlah hanya *keyword* yang memiliki kata yang tidak berlebihan.
2. Jika pada algoritma *string matching* dapat mencocokkan kata atau *keyword* yang *related* atau sesuai dengan apa yang dicari, maka pada *regular expression* (regex) dapat berfungsi dalam pencarian yang lebih kompleks dengan menggunakan pola.
3. Pada kamus besar bahasa indonesia (KBBI) *string matching* cocok hanya jika *string* yang dicari secara keseluruhan ada dalam kamus sedangkan untuk *regular expression* dapat berfungsi dalam pencarian pola spesifik, seperti mencari kata yang diawali atau diakhiri oleh karakter tertentu, kata dengan panjang tertentu, kata dengan kombinasi huruf tertentu, dan sebagainya.

REFERENSI

- [1] A. D. Hartanto, N. Norhikmah, D. Prabowo, I. N. Fajri, W. Widayani, and others, “Implementation of the Levenshtein Distance Algorithm and the Regular Search Expression Method for Detecting Typors in Javascript,” *Sistemasi: Jurnal Sistem Informasi*, vol. 12, no. 2, pp. 452–464, 2023.
- [2] A. S. Sumi, P. Purnawansyah, and L. Syafie, “Analisa Penerapan Algoritma Brute Force Dalam Pencocokan String,” in *Prosiding SAKTI (Seminar Ilmu Komputer dan Teknologi Informasi)*, 2018, pp. 88–92.
- [3] D. Abdurahman and I. Kurniawan, “Rancang Bangun Aplikasi Kamus Fisika Dasar Menggunakan Algoritma String Matching Brute Force Berbasis Android,” in *Seminar Nasional Teknologi Informasi*, 2018, pp. 150–155.
- [4] A. Januardi, “Analisa Perbandingan Algoritma Brute Force dan Boyer Moore dalam Pencarian Word Suggestion Menggunakan Metode Perbandingan Eksponensial,” *STMIK Budi Darma Medan*, 2013.
- [5] I. Mahdalena, “IMPLEMENTASI ALGORITMA STRING MATCHING PADA KAMUS ISTILAH AKUNTANSI BERBASIS ANDROID,” *JURIKOM (Jurnal Riset Komputer)*, vol. 4, no. 4, 2017.
- [6] R. T. Handayanto, A. W. Amrillah, I. Juwita, M. Arifin, S. Setiaji, and R. Karani, “Algoritma Stemming Sebagai Pra-Proses Pengecekan Kemiripan Naskah,” *Bina Insani ICT Journal*, vol. 5, no. 2, pp. 175–182, 2018.
- [7] F. T. Waruwu and P. Ramadhani, “Perancangan Aplikasi Translator Bahasa Indonesia Ke Bahasa Inggris ESP (English For Specific Purpose) Komputer Sains Untuk Peningkatan Informasi Mahasiswa Menggunakan Brute Force Method,” *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 2, no. 1, 2018.