

IMPLEMENTASI ALGORITMA AHO-CORASICK PADA PENCARIAN ARTI ISTILAH BAHASA GAUL

Nuraisyah¹, Gudio Leonarde Ginting², Edizal Hatmi³

STMIK Budi Darma, Medan, Indonesia
Jl. Sisingamangaraja No. 338 Medan

ABSTRAK

Di tahun 2001, banyak kelompok yang mengembangkan bahasa-bahasa gaul dan kemudian menciptakan bahasa baru. Hal ini sengaja dilakukan agar orang lain tidak mengerti apa-apa yang mereka bicarakan. Tekadang bahasa gaul diciptakan pula dari kata-kata aneh yang sulit dilacak asal mulanya yang membuat pendengar sulit untuk memahaminya. Untuk lebih mudah dalam memahami arti dari istilah bahasa gaul dalam kehidupan sehari-hari, maka dilakukan suatu pencarian, pencarian merupakan sebuah permasalahan yang umum terjadi terutama dalam dunia komputer. Terdapat banyak algoritma untuk menyelesaikan permasalahan ini dan tentunya dengan kompleksitas yang berbeda-beda. Termasuk juga permasalahan dalam mencari kata dalam sebuah kumpulan kata atau kalimat. Pada penelitian ini algoritma yang digunakan yaitu algoritma aho-corasick. Berdasarkan penelitian Khan pada tahun 2012, Algoritma Aho-Corasick terbukti lebih cepat dibandingkan algoritma lainnya dalam hal mencari kata. Penelitian ini bertujuan untuk mengimplementasikan algoritma Aho-Corasick pada pencarian arti dari istilah bahasa gaul. Penyelesaian ini dilakukan secara tahap demi tahap dan menyediakan fasilitas pencarian sekumpulan pola pada sekumpulan file yang terdapat pada lokasi pencarian tertentu.

Kata Kunci: Bahasa Gaul, Pencarian, Algoritma, Aho-Corasick

I. PENDAHULUAN

Di tahun 2001, banyak kelompok yang mengembangkan bahasa-bahasa gaul ini dan kemudian menciptakan bahasa baru. Hal ini sengaja dilakukan agar orang lain tidak mengerti apa-apa yang mereka bicarakan, agar mereka dapat dengan bebas membicarakan hal yang bersifat sensitif. Tekadang bahasa gaul diciptakan pula dari kata-kata aneh yang sulit dilacak asal mulanya. Kalimat-kalimat yang digunakan umumnya kalimat tunggal. Bentuk-bentuk elip juga banyak digunakan untuk membuat susunan kalimat menjadi lebih pendek sehingga seringkali dijumpai kalimat-kalimat yang tidak lengkap. Dengan menggunakan struktur yang pendek, pengungkapan makna menjadi lebih cepat yang sering membuat pendengar yang bukan penutur asli bahasa Indonesia mengalami kesulitan untuk memahaminya. Maka dari itu untuk lebih mudah dalam pemahaman tentang arti dari istilah bahasa gaul dalam kehidupan sehari-hari, maka dibutuhkan suatu perancangan sistem yang menggunakan suatu algoritma yang dapat mempermudah untuk pencarian arti dari istilah bahasa gaul.

Salah satu algoritma yang digunakan untuk perancangan sistem pencarian adalah algoritma aho-corasick. Beberapa penelitian terdahulu yang menggunakan algoritma aho-corasick yaitu penelitian Tiara prawisuda yang berjudul implementasi algoritma aho-corasick pada plugin wordpress penghitung kepadatan kata kunci dan penelitian Andrew handoko yang berjudul file undelete untuk memulihkan file yang telah terhapus dari file system dengan algoritma aho-corasick.

Algoritma aho carasick adalah algoritma yang dapat digunakan untuk mencari dimana sebuah string (dalam kasus ini dinamakan sebuah pola) apakah ditemukan di dalam kumpulan string lain dengan ukuran yang lebih besar. Algoritma aho-corasick ini

merupakan algoritma berwaktu linier yang didasari pada pendekatan automata dan hanya dapat bekerja optimal pada kasus terburuk (*worst case*). Dalam segi efisiensi waktu, algoritma aho-corasick berbanding terbalik dengan algoritma boyer moore. Algoritma boyer moore merupakan salah satu *single pattern matching*. Algoritma ini beranggapan bahwa terdapat kemungkinan untuk mengabaikan sekumpulan teks dengan porsi besar pada saat pencarian, sehingga mampu memberikan waktu eksekusi yang lebih cepat dari pada algoritma linier pada kasus rata-rata (*average case*) [1].

II. TEORITIS

A. Implementasi

Menurut Charles O Jones Implementasi sebagai “a process of getting additional resources so as to be figure out of to be done”. Implementasi dalam hal ini diartikan sebagai suatu proses mendapatkan sumber daya tambahan, dapat menghitung apa yang dapat dikerjakan [2].

implementasi adalah “proses penerapan ide, konsep, kebijakan, atau inovasi dalam suatu tindak praktis sehingga memberikan dampak baik berupa perubahan pengetahuan, keterampilan, maupun nilai dan sikap”. Penjelasan tersebut dapat dipahami bahwa implementasi merupakan hasil [3].

Berdasarkan uraian diatas dapat disimpulkan bahwa implementasi merupakan suatu proses atau rangkaian kegiatan dalam rangka pelaksanaan kebijakan dengan melibatkan sumberdaya yang ada, sarana dan prasarana untuk diterapkan kepada masyarakat guna mencapai tujuan yang diinginkan oleh pembuat kebijakan.

B. Bahasa Gaul

Bahasa gaul adalah gaya bahasa yang merupakan perkembangan atau modifikasi dari

berbagai macam bahasa, termasuk bahasa Indonesia sehingga bahasa gaul tidak memiliki sebuah struktur gaya bahasa yang pasti. Sebagian besar kata-kata dalam bahasa gaul remaja merupakan terjemahan, singkatan, maupun pelesetan. Namun, terkadang diciptakan pula kata-kata aneh yang sulit dilacak asal mulanya. Kalimat-kalimat yang digunakan umumnya kalimat tunggal. Bentuk-bentuk elip juga banyak digunakan untuk membuat susunan kalimat menjadi lebih pendek sehingga seringkali dijumpai kalimat-kalimat yang tidak lengkap. Dengan menggunakan struktur yang pendek, pengungkapan makna menjadi lebih cepat yang sering membuat pendengar yang bukan penutur asli bahasa Indonesia mengalami kesulitan untuk memahaminya [4].

C. Struktur Bahasa Gaul

Ragam bahasa gaul remaja memiliki ciri khusus, singkat, lincah dan kreatif. Katakata yang digunakan cenderung pendek, sementara kata yang agak panjang akan diperpendek melalui proses morfologi atau menggantinya dengan katayang lebih pendek seperti “memang menjadi emang”. Kalimat-kalimat yang digunakan kebanyakan berstruktur kalimat tunggal. Bentuk-bentuk elip juga banyak digunakan untuk membuat susunan kalimat menjadi lebih pendek sehingga seringkali dijumpai kalimat-kalimat yang tidak lengkap. Dengan menggunakan struktur yang pendek, pengungkapan makna menjadi lebih cepat yang sering membuat pendengar yang bukan penutur asli bahasa Indonesia mengalami kesulitan untuk memahaminya.

a. Penggunaan awalan e

Kata “emang” itu bentukan dari kata “memang” yang disisipi bunyi e. Disini jelas terjadi pemendekan kata berupa mengilangkan huruf depan (m). Sehingga terjadi perbedaan saat melafalkan kata tersebut dan merancu dari kata aslinya.

b. Kombinasi k, a, g

Kata “kagak” bentukan dari kata “tidak” yang bunyinya tiddiganti kag. Huruf konsonan pada kata pertama diganti dengan k huruf vocal i diganti a. Huruf konsonan kedua diganti g. sehingga kata tidak menjadi kagak.

c. Sisipan e

Kata “temen” merupakan bentukan dari kata “teman” yang huruf vokal a menjadi e. Hal ini mengakibatkan terjadinya perbedaan pelafalan.

C. Algoritma Aho-Corasick

Aho-Corasick adalah salah satu algoritma pencocokan *string*, yang diiciptakan oleh Alfred V. Aho dan Margaret J. Corasick. Untuk menemukan kejadian *multipattern* pada *string* teks, algoritma ini lebih tepat karena dapat melakukan pencocokan tepat dari pola dalam teks [5].

Algoritma Aho-Corasick adalah pengembangan dari algoritma Knuth-Morris-Pratt, yaitu algoritma pertama yang memecahkan berbagai pola pencocokan

pola multipel dalam waktu linier berdasarkan pendekatan automata. Algoritma ini berfungsi sebagai basis tool figur UNIX. Algoritma Aho-Corasick terdiri dari dua bagian. Pada bagian pertama mereka membuat mesin pencocokan pola hingga set dari kata kunci dan pada bagian kedua, string teks sebagai masukan diterapkan pada pola pencocokan string.

Kebutuhan ruang atau memori dari algoritma Aho-Corasick bisa sangat besar tergantung pada pola dan juga panjang masing-masing pola dalam set pola proses pencocokan hanya melangkah melalui input karakter satu per satu dan memeriksa apakah ada yang cocok. Setiap langkah dalam pencocokan pola mesin terjadi dalam waktu konstan. Jadi, matcher Aho-Corasick selalu beroperasi dalam $O(n)$ running time.

III. ANALISA DAN PEMBAHASAN

Data yang digunakan dalam penelitian ini Pencarian istilah bahasa gaul yang telah ada saat ini pada umumnya hanya menggunakan metode pencarian yang hanya mendukung problema *single pattern matching* atau *string matching* biasa. Penggunaan algoritma *single pattern matching* atau *string matching* biasa tidak efisien apabila diterapkan untuk mencari beberapa kata sekaligus dan seringkali terdapat kemungkinan untuk mengabaikan sekumpulan kata dengan porsi besar pada saat pencarian.

Pada sistem ini, aplikasi akan menggunakan algoritma aho-corasick yang dapat digunakan pada problema *multi pattern matching* yang dapat mendukung pencarian banyak kata, sehingga hasil yang diperoleh akan jauh lebih akurat. Selain itu, algoritma *multi pattern matching* juga merupakan algoritma yang berkonsentrasi pada pencarian biasa dari pada keadaan kasus terburuk (*worst case*), sehingga diyakini bahwa algoritma ini jauh lebih efisien dari pada algoritma lainnya pada implementasinya.

Contoh:

Teks : KAMSEUPAY, MACAPAH, CIUS MIAPAH
Patern : PAH

Cara kerja:

| Langkah Ke-1 | | | | | | | | | | |
|--------------|---|---|---|---|---|---|---|---|---|----|
| Patern | P | A | H | | | | | | | |
| Teks 1 | K | A | M | S | E | U | P | A | Y | |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| Teks 2 | M | A | C | A | P | A | H | | | |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | |
| Teks 3 | C | I | U | S | | M | I | A | P | A |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Langkah 1: dalam proses pencocokan pertama tidak ditemukan kemiripan *pattern* dengan *teks* maka selanjutnya akan dilakukan pergeseran satu karakter ke kanan untuk pencocokan *string* berikutnya

| Langkah Ke-2 | | | | | | | | | | |
|--------------|---|---|---|---|---|---|---|---|---|----|
| Patern | | P | A | H | | | | | | |
| Teks 1 | K | A | M | S | E | U | P | A | Y | |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| Teks 2 | M | A | C | A | P | A | H | | | |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | |
| Teks 3 | C | I | U | S | | M | I | A | P | H |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Langkah 2: dalam proses pencocokan pertama tidak ditemukan kemiripan *pattern* dengan *teks* maka selanjutnya akan dilakukan pergeseran satu karakter ke kanan untuk pencocokan *string* berikutnya

| Langkah Ke-3 | | | | | | | | | | |
|--------------|---|---|---|---|---|---|---|---|---|----|
| Patern | | | P | A | H | | | | | |
| Teks 1 | K | A | M | S | E | U | P | A | Y | |
| indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| Teks 2 | M | A | C | A | P | A | H | | | |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | |
| Teks 3 | C | I | U | S | | M | I | A | P | H |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Langkah 3: dalam proses pencocokan pertama tidak ditemukan kemiripan *pattern* dengan *teks* maka selanjutnya akan dilakukan pergeseran satu karakter ke kanan untuk pencocokan *string* berikutnya.

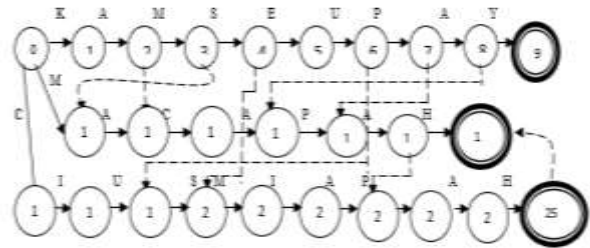
| Langkah Ke-4 | | | | | | | | | | |
|--------------|---|---|---|---|---|---|---|---|---|----|
| Patern | | | | P | A | H | | | | |
| Teks 1 | K | A | M | S | E | U | P | A | Y | |
| indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |
| Teks 2 | M | A | C | A | P | A | H | | | |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | |
| Teks 3 | C | I | U | S | | M | I | A | P | H |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Langkah 4: dalam proses pencocokan pertama tidak ditemukan kemiripan *pattern* dengan *teks* maka selanjutnya akan dilakukan pergeseran satu karakter ke kanan untuk pencocokan *string* berikutnya.

| Langkah Ke-5 | | | | | | | | | | |
|--------------|---|---|---|---|---|---|---|---|---|--|
| Patern | | | | | P | A | H | | | |
| Teks 1 | K | A | M | S | E | U | P | A | Y | |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | |

| | | | | | | | | | | | |
|--------|---|---|---|---|---|---|---|---|---|----|----|
| Teks 2 | M | A | C | A | P | A | H | | | | |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | | |
| Teks 3 | C | I | U | S | | M | I | A | P | A | H |
| Indeks | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |

Langkah 5: dan pada proses pencocokan kelima akhirnya ditemukan kecocokan *pattern* pada teks MACAPAH

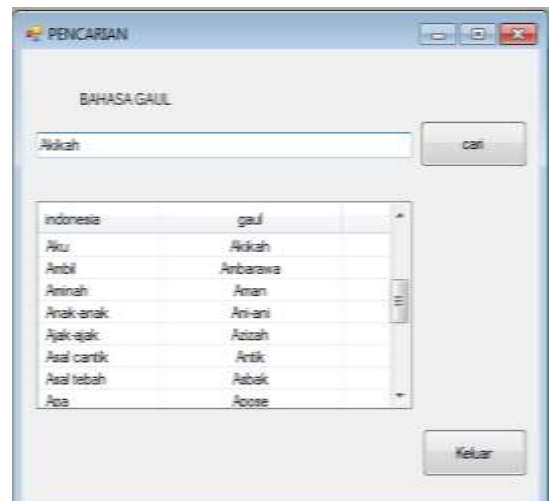


Gambar 3. Automata Macapah

IV. IMPLEMENTASI

Implementasi merupakan langkah yang digunakan untuk mengoperasikan sistem yang dibangun. Dalam bab ini dijelaskan bagaimana menjalankan sistem tersebut. Sistem pengolahan program merupakan suatu kesatuan pengolahan yang terdiri dari prosedur dan pelaksanaan data. Komputer sebagai sarana pengolahan program harus menyediakan fasilitas-fasilitas pendukung dalam pengolahan nantinya.

Tampilan berikut ini adalah form pencarian dimana dapat berfungsi untuk mencari bahasa gaul serta artinya.



Gambar 4. pencarian bahasa gaul

V. KESIMPULAN

Berdasarkan hasil penelitian analisa dan perancangan aplikasi arti istilah bahasa bahasa gaul ini maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Perangkat lunak pencarian dapat membantu proses pencarian kata bahasa bahasa gaul. Proses pencarian kata lebih mudah dan cepat dibandingkan

dengan menggunakan kamus dalam bentuk buku. Jumlah kata yang dalam pencarian dapat ditambah oleh Admin.

2. Penerapan algoritma aho-corasick dalam pencarian arti istilah bahasa gaul memberikan performansi baik (terutama dari segi kecepatan) yang dapat membantu pengguna dalam mencari kata istilah sehingga layak untuk dipakai.
3. Bahasa pemrograman Microsoft Visual Basic dapat membangun berbagai aplikasi yang dapat digunakan sebagai sarana pendidikan.

REFERENCES

- [1] Adi Sutojo, "Pengaruh Pelaksanaan Kebijakan Atau Aturan Terhadap Pelayanan Sektor Publik Di Kelurahan Purwodadi Kecamatan Arga Makmur Kabupaten Bengkulu Utara," *Jurnal Penelitian Sosial Dan Politik*, Vol. 4, Pp. 1-11, 2015.
- [2] Edi Wijawa, "Analisis Penggunaan Algoritma *Breadth First Search* Dalam Konsep *Artificial Intellegencia*," *Jurnal Time*, Vol. ii, Pp. 18-26, 2013.
- [3] Ratna Prasasti Suminar, "Pengaruh Bahasa Gaul Terhadap Penggunaan Bahasa Indonesia," *Jurnal Logika*, Vol. Xviii, Pp. 114-119, 2016.
- [4] Panny Rahayuningsih Agustia, "Analisis Perbandingan Kompleksitas Algoritma Pengurutan Nilai (Sorting)," *Jurnal Evolusi*, Vol. 4, Pp. 64-75, 2016.
- [5] Saima Hasib, Mahak Motwani, And Amit Saxena, "Importance Of Aho-Corasick String Matching Algorithm In Real World Applications," *Journal Of Computer Science And Information Technologies*, Vol. 4, Pp. 467-469, 2013.
- [6] A. Fau, M. Mesran, and G. L. Ginting, "Analisa Perbandingan Boyer Moore Dan Knuth Morris Pratt Dalam Pencarian Judul Buku Menerapkan Metode Perbandingan Eksponensial (Studi Kasus : Perpustakaan STMIK Budi Dharma)," *J. Times (Technology Informatics Comput. Syst.*, vol. 6, no. 1, pp. 12-22, 2017.
- [7] M. Mesran, "IMPLEMENTASI ALGORITMA BRUTE FORCE DALAM PENCARIAN DATA KATALOG BUKU PERPUSTAKAAN," *Maj. Ilm. INTI*, vol. 3, no. 1, pp. 100-104, 2014.
- [8] J. I. Sinaga, Mesran, and E. Buulolo, "APLIKASI MOBILE PENCARIAN KATA PADA ARTI AYAT AL-QUR'AN BERBASIS ANDROID MENGGUNAKAN ALGORITMA STRING MATCHING," *INFOTEK*, vol. 2, no. 2, pp. 68-72, 2016.
- [9] F. T. Waruwu and M. Mesran, "IMPLEMENTASI ALGORITMA KNUTH MORRIS PRATT PADA APLIKASI KAMUS ISTILAH LATIN FLORA DAN FAUNA BERBASIS ANDROID," *Maj. Ilm. INTI*, vol. 4, no. 1, pp. 96-102, 2014.