

Penerapan Algoritma Zhutakaoka Pada Aplikasi Pengenalan Rambu K3

Evi Ramadani

Program Studi Teknik Informatika Universitas Budi Dharma, Medan, Indonesia

Email: Eviramadani049@gmail.com

Abstrak—Rambu k3 merupakan rambu yang sangat penting dalam pekerjaan untuk menghindari kecelakaan kerja dan rambu-rambu k3 ini sangat banyak macam seperti gambar, dan lambang sehingga rambu-rambu k3 ini sangat bermanfaat untuk para pekerja baik di bangunan, atau di proyek. Agar para pekerja terhindar dari kecelakaan kerja baik di lingkungan kerja atau di luar lingkungan kerja. Untuk itu diperlukan suatu Aplikasi yang memuat tentang pencarian rambu k3 dimana pekerja dapat mempelajari rambu tersebut. Aplikasi rambu k3 ini digunakan pekerja dengan mencari dari suatu Aplikasi yang ada pada Aplikasi yang telah dibuat. Dan dapat dipahami apabila pekerja berhasil mencari rambu-rambu k3 yang telah di tentukan sesuai dengan pattern yang ada, tetap pencarian solusi pada Aplikasi. Pencarian rambu k3 di rancang dengan menggunakan Metode String matching, pengembangan Aplikasi menggunakan metode perancangan UML dan dibangun pada Platform android.

Kata Kunci: String Matching; Rambu K3; Algoritma Zhutakaoka

Abstract—K3 signs are very important signs in work to avoid work accidents and there are many kinds of K3 signs such as pictures and symbols so that these K3 signs are very useful for workers both in buildings and in projects. So that workers avoid work accidents both in the work environment or outside the work environment. For that we need an application that contains the search for K3 signs. Where workers can learn these signs. This K3 sign application is used by workers by searching from an existing application in the application that has been made. And it can be understood if the worker succeeds in looking for the K3 signs that have been determined according to the existing pattern, still searching for solutions in the application. Search for K3 signs is designed using the String matching method, application development uses the UML design method and is built on the Android platform.

Keywords: String Matching; K3 Signs; Zhutakaoka Algorithm

1. PENDAHULUAN

K3 merupakan singkatan dari keselamatan dan kesehatan kerja. Rambu K3 sering dijumpai pada proyek, atau bangunan. Keselamatan dan kesehatan kerja (K3) diartikan suatu perlindungan kepada tenaga kerja, barang atau alat agar selalu dalam keadaan selamat dan sehat selama berada di tempat kerja. Banyak pekerja yang mengalami kecelakaan dalam bekerja, dan dapat terjadi kapan pun dimanapun. Kecelakaan kerja tidak saja menimbulkan korban jiwa tetapi juga menimbulkan kerugian bagi setiap pekerja, dan dapat mengganggu proses pekerjaan secara menyeluruh baik diri sendiri atau masyarakat yang lain.

Dan pentingnya kita mengetahui mengenai K3 agar setiap pekerja mendapat jaminan keselamatan dan kesehatan kerja baik secara fisik, sosial, dan psikologis, untuk setiap pekerja merasa aman dan terlindungi dengan adanya jaminan keselamatan dan kesehatan kerja, pekerja juga terlindungi dari bahaya akibat kerja dan lingkungan kerja. Maka untuk mempermudah pekerja mempelajari rambu diperlukan pengembangan sistem dengan merancang aplikasi pengenalan rambu K3 yang memiliki fitur pencarian yang baik guna meminimalisir waktu pekerja untuk menemukan hasil pencarian rambu K3 yang akan dipelajari. Dan dengan adanya aplikasi ini dapat membantu pekerja menghindari terjadinya kecelakaan di dalam atau di luar tempat kerja. Dengan menggunakan aplikasi ini pekerja bisa selalu hati-hati dalam melakukan pekerjaan.

2. METODOLOGI PENELITIAN

2.1 Algoritma Zhutakaoka

Algoritma Zhutakaoka diciptakan oleh R F Zhu dan T. Takaoka pada tahun 1989. Algoritma ini menggunakan dua text karakter untuk menghitung pergeseran karakter berdasarkan *bad-character*. untuk mencari ketidak cocokan atau mencari seluruh text menggunakan *processing hashing*. algoritma Zhutakaoka menjelaskan bahwa pencarian pattern mulai dari tiap baris mulai dari baris ke 0 dan berakhir pada baris ke $nl - ml$.

Zhu dan takaoka merancang sebuah algoritma yang dapat bekerja menggunakan pergeseran *bad-character* untuk dua karakter text secara berturut-turut. selama fase pencarian kecocokan karakter di lakukan dari kanan ke kiri dan ketika proses berada di posisi $y[j...j + m - 1]$ dan ketidak cocokan terjadi di antara $x[m - k]$ dan $y[j + m - k]$ ketika $x[m - k + 1..m - 1] = y[j + m - k + 1..j + m - 1]$ pergeseran dilakukan dengan pergeseran *bad-character* untuk text karakter $y[j + m - 2]$ dan $y[j + m - 1]$. tabel good-suffix dari algoritma Boyer-Moore juga digunakan untuk menghitung persamaan [1], [2].

2.2 Rambu Keselamatan Dan Kesehatan kerja

Rambu keselamatan dan kesehatan kerja adalah untuk melindungi pekerja terhadap potensi yang membahayakan dan menimbulkan penyakit akibat pekerjaan, pencegahan kecelakaan penerasian peralatan kerja, dan sifat pekerja serta

masyarakat sekitar, tujuan agar pekerja mempunyai ketahanan fisik, daya kerja, dan tingkat kesehatan yang tinggi sehingga tercipta situasi kerja yang nyaman dengan dengan tingkat keselamatan kerja yang tinggi [3].

Rambu merupakan alat bantu menginformasikan dan untuk melindungi keselamatan dan kesehatan para masyarakat yang berada di lingkungan kerja. rambu juga memuat salah satu alat pelengkap dalam bentuk tertentu lambang, huruf, angka, kalimat atau panduan. Yang di gunakan untuk memberikan peringatan, larangan, perintah dan petunjuk bagi masyarakat[4]. Dan yang harus kita pelajari dari rambu K3 di antaranya warna kuning menginformasikan untuk potensi bahaya. dan warna merah menginformasikan untuk larangan pemadaman api, warna hijau menginformasikan untuk zona aman pertolongan, warna biru menginformasikan untuk wajib ditaati, dan warna putih menginformasikan untuk informasi umum.

3. HASIL DAN PEMBAHASAN

Sebagai contoh akan dilakukan proses pencarian *pattern* menggunakan algoritma *Zhutaoka* dengan pergeseran dari kanan ke kiri. Text : “BAHAYA MEDAN MAGNET” dan *pattern*:” MAGNET”

Langkah-langkah algoritma yaitu:

- a. Langkah awal adalah menentukan table *bmBc*. Untuk memudahkan menentukan table *ztBc* adalah dengan menemukan tabel *bmBc* untuk mengidentifikasi setiap perpindahan. Dan dengan rumus :

M-1- i

Tabel 1. *bmBc*

Index	0	1	2	3	4	5
Karakter	M	A	G	N	E	T
Nilai OH	5	4	3	2	1	0

- b. Langkah selanjutnya adalah masukkan semua karakter yang terdapat pada sumber teks dan buatlah menjadi dua dimensi, karakter yang dimasukkan di urutan sesuai abjad tak berulang.

Tabel 2. *ztBc*

	A	E	G	M	N	T	*
A							
E							
G							
M							
N							
T							
*							

- c. langkah selanjutnya adalah mengisi tabel dengan cara melihat karakter pada baris dan karakter pada kolom. langkah untuk mencari tabel *ZtBc* yaitu:

1. Jika pola yang dicari ada, maka masukkan nilai OH ke karakter kolom.
2. Jika pola yang dicari tidak ada maka masukkan panjang *pattern*.
3. Jika karakter pada suatu kolom yang di tuju adalah karakter pertama pada *pattern*, maka masukkan nilai OH karakter tersebut, tanpa melihat karakter baris.

Tabel 3. *ztBc*

	A	E	G	M	N	T	*
A	6	6	6	5	6	6	6
E	6	6	1	5	2	6	6
G	6	3	6	5	6	6	6
M	4	6	6	5	6	5	6
N	6	6	6	5	6	6	6
T	6	6	6	5	6	6	6
*	6	6	6	5	6	6	6

- d. Setelah mendapatkan hasil akhir pada tabel *ztBc*, maka dilanjutkan dengan membuat tabel *suffix* dari *pattern*. Berikut adalah tabel *zmGs*.

Tabel 4. *zmGs*

Index	0	1	2	3	4	5
Karakter	M	A	G	N	E	T
Nilai MH	6	6	6	6	6	1

- e. Langkah terakhir adalah melakukan pencocokan *pattern* dengan sumber teks berikut:

Teks : BAHAYA MEDAN MAGNET

Pattern : MAGNET

Tabel 5. Percobaan ke-1

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Teks	B	A	H	A	Y	A		M	E	D	A	N		M	A	G	N	E	T
Pattern	M	A	G	N	E	T													

Pada percobaan ke-1 dapat dilihat karakter A dengan T tidak sama dengan karakter *pattern* yang lainnya, sehingga di lakukan pergeseran sebanyak jumlah karakter yaitu 4 langkah.

Table 6. Percobaan ke-2

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Teks	B	A	H	A	Y	A		M	E	D	A	N		M	A	G	N	E	T
Pattern						M	A	G	N	E	T								

Pada percobaan ke-2 dapat dilihat karakter D dengan T tidak sama dengan karakter *pattern* yang lainnya, sehingga di lakukan pergeseran sebanyak jumlah karakter yaitu 6 langkah.

Tabel 7. Percobaan ke-3

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Teks	B	A	H	A	Y	A		M	E	D	A	N		M	A	G	N	E	T
Pattern											M	A	G	N	E	T			

Pada percobaan ke-3 dapat dilihat karakter G dengan T tidak sama dengan karakter *pattern* yang lainnya, sehingga di lakukan pergeseran sebanyak jumlah karakter yaitu 3 langkah.

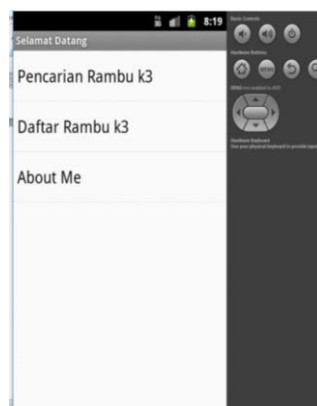
Tabel 8. Percobaan ke-4

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Teks	B	A	H	A	Y	A		M	E	D	A	N		M	A	G	N	E	T
Pattern														M	A	G	N	E	T

Pada percobaan ke-4 dapat dilihat bahwa karakter T sejajar dengan T, memiliki kesamaan dengan karakter *pattern* yang lainnya, sehingga pergeseran karakter *pattern* berhenti.

3.1 Implementasi Program

Untuk menguji Aplikasi *Eclips juno* Rambu k3, maka perlu mempersiapkan sistem yang terdiri dari *hardware*(perangkat keras), dan *software*(perangkat lunak). Saat pertama kali Ditampilkan akan muncul Menu Utama:



Gambar 1. Menu Utama

Menu utama merupakan media penghubung pengguna aplikasi dengan aplikasi Rambu k3 yang dibangun oleh penulis pada penelitian ini untuk memilih menu yang ada pada aplikasi Rambu k3 yang dibangun.



Gambar 2. Pencarian Rambu k3

Rambu k3 merupakan media penghubung pengguna aplikasi dengan aplikasi pencarian Rambu k3 yang di bangun oleh penulis pada penelitian ini untuk menampilkan proses pencarian Rambu k3.



Gambar 3. Daftar Rambu k3

Daftar Rambu k3 merupakan media penghubung pengguna aplikasi dengan aplikasi pencarian rambu k3 yang digunakan penulis pada penelitian ini untuk menampilkan Daftar rambu- rambu k3 yang di cari setelah melakukan pencarian rambu k3.

4. KESIMPULAN

Berdasarkan hasil penelitian dapat penulis simpulkan bahwa aplikasi pencarian arti pada pengenalan rambu k3 dilakukan untuk mempercepat user dalam pencarian rambu k3. Implementasi algoritma *zhu takaoka* dapat mempermudah proses pencarian isi Rambu k3. Aplikasi Rambu k3 pada penelitian ini dibuat menggunakan editor *eclipse juno*.

REFERENCES

- [1] G. Togarotop, A. Erlansari, and F. F. Coastera, "Implementasi Algoritma Zhu-Tajkaoka Pada Aplikasi Kamus Istilah Musik Berbasis Android," vol. 5, no. 2, pp. 147–153, 2017.
- [2] A. Kusnadi and A. K. Wicaksono, "Perbandingan Algoritma Horspool dan Algoritma Zhu-Takaoka dalam Pencarian String Berbasis Desktop," *J. Ultim. Comput.*, vol. 9, no. 1, pp. 12–16, 2018.
- [3] mP PROF.dR.iR.hARTUTIK, *METODE ANALISA MUTU PAKAN*. malang: eLEKTRONIK PERTAMA DAN TERBESAR DI INDONESIA, 2012.
- [4] H. & Haikal, *seluk beluk pedagang ekspor & impor*, BUSHINDO. jakarta timur, 2012.