Seminar Nasional Sains & Teknologi Informasi (SENSASI)
SENSASI 2019

ISBN: 978-602-52720-2-8
Juli 2019
Hal: 609 – 613

Implementasi Algoritma Zhu-Takaoka Pada Pencarian Data Kependudukan Berbasis Android

Annisah, Indah Permata Sari, Konisius Sihura, Nelly Astuti Hasibuan

Program Studi Teknik Informatik, STMIK Budi Darma, Medan, Indonesia
Jalan Sisingamangaraja No. 338, Medan, Indonesia
Email: [1]annisah.annisah123@email.com, [2]indahmatasari19@email.com , [3]konisiussihura21@gmail.com

Abstrak

KTP adalah indentitas resmi penduduk sebagai butki diri yang diterbitkan oleh instansi pelaksanaan yang berlaku diseluruh wilayah negara Indonesia, kartu ini wajib dimiliki setiap warga Negara Indonesia yang sudah berumur 17 tahun. Namun permasalah yang ada saat ini adalah terdapat kendala pada pencarian identitas kependudukan. Dimana banyak masyarakat yang terkadang lupa membawa identitas diri, sehingga dengan adanya kendala ini harus ada *tools* atau alat bantu yang memudahkanya dalam pencarian identitas. Dalam penyelesaian tersebut penulis menggunakan metode *String matching* kami menggunakan algoritma *Zhu-Takaoka* dimana *Zhu-Takaoka* menjelaskan bahwa pencarian pattern dilakukan mulai dari tiap baris, mulai dari baris ke-0 dan berakhir pada baris ke nl-ml. *Zhu* dan *Takaoka* merancang sebuah algoritma yang cara kerjanya menggunakan pergeseran dengan *bad-character* dua karakter teks secara untuk berturut-turut.

Kata Kunci: *string matching, string, zhu-takaoka, pattern, text.*

1. PENDAHULUAN

Indonesia adalah Negara yang memiliki jumlah penduduk terbesar ke 4 didunia. Selain jumlah penduduk indonesia yang besar, luasnya Negara Indonesia yang memiliki banyak pulau semakin Indonesia banyak mengalami permasalahan kependudukan. Kemajuan teknologi saat ini tidak dapat kita hindari, karena kemajuan teknologi akan terus berkembang seiring dengan perkembangan ilmu pengetahuan. Namun masalah pada saat ini terdapat kendala pada pencarian identitas kependudukan. Dimana banyak masyarakat yang terkadang lupa membawa identitas diri, sehingga dengan adanya kendala ini harus ada *tools* atau alat bantu yang memudahkanya dalam pencarian identitas.

String matching adalah metode yang digunakan untuk mengatasi permasalahan diatas, *String matching* merupakan suatu metode yang dapat digunakan untuk menemukan keakuratan atau hasil dari suatu atau teks yang diberikan. *String matching* merupakan pokok bahasan yang penting dalam ilmu komputer karna teks merupakan bentuk utama dari pertukaran informasi antar manusia. Dalam penerapan metode *String matching* diperlukan sebuah media yang dapat memfasilitasi pergerakan dan akses informasi yang cepat, dimanapun dan kapan saja. Mengingat masyarakat saat ini akrab dengan *smartphone*, dengan ini kami mengimplementasikan metode *String matching* dengan *Android*. *Android* adalah salah satu sistem operasi *smartphone* yang telah berkembang di tengah-tengah masyarakat. Adapun keunggulan dari sistem opersi ini antar lain ialah sistem opersinya yang terbuka (*open source*).

Dengan menggunakan metode *String matching* kami menggunakan algoritma *Zhu-Takaoka* dimana *Zhu-Takaoka* menjelaskan bahwa pencarian pattern dilakukan mulai dari tiap baris, mulai dari baris ke-0 dan berakhir pada baris ke nl-ml. *Zhu* dan *Takaoka* merancang sebuah algoritma yang cara kerjanya menggunakan pergeseran dengan bad-character untuk dua karakter teks secara berturut-turut. Dengan adanya sistem operasi berbasis android, diharapkan dapat mempermudah masyarakat dalam melakukan pencarian dan melihat identitas diri.

2. TEORITIS

2.1 String Matching

String Matching merupakan proses pencarian semua kemunculan queri yang selanjutnya di sebut pattern ke dalam string yang lebih panjang atau teks. Pencocokan string adalah bagian terpenting proses pencarian string dalam dokumen yang berkaitan dengan teknik dan cara pencocokan string yang sering di gunakan. Berikut adalah prinsip cara kerja algoritma string matching:

1. Mengeserkan teks dengan bantuan window yang ukurannya sama dengan pattern.
2. Menempatkan window keawal teks.
3. Membandingkan karakter window dengan karakter pattern. Setelah pencocokan selesai, maka di lakukan shift ke kanan pada window. pencocokan ini di lakukan berulang-ulang sampai window berada pada akhir teks. Mekanisme ini di sebut mekanisme suding window.

2.2 Algoritma Zhu-Takaoka

BM" Algoritma (Algoritma Zhu-Takaoka) merupakan lanjutan dari algoritma pencocokan string *Boyer-Moore Algorithm* yang diciptakan oleh Boyer R.S dan Moore J.S tahun 1997. Algoritma ini dianggap sebagai algoritma yang paling efisien pada aplikasi umum. Algoritma Zhu Takaoka diperkenalkan oleh Zhu Rui Feng dan Tadao Takaoka pada tahun 1986. Dalam makalahnya, Zhu dan Takaoka menyebut algoritma pencocokan string ini sebaga BM" Algorithm (Boyer-Moore" Algorithm). Algoritma *Zhu-Takaoka* dan algoritma *Boyer Moore* memiliki ciri sama dalam proses melakukan pencarian *String*, yaitu terdapat tahap *Preprocessing*, *Right-to-left scan*, *Bad character* dan *Good-suffix rule*. Sementara itu, perbedaan dari dua algoritma tersebut terletak pada tahap penentuan I*Bad character rule*. Dalam *Boyer-Moore*, *bad*

ISBN: 978-602-52720-2-8
Juli 2019
Hal: 609 – 613

character hanya terdiri dari *array* satu dimensi, sedangkan dalam *Zhu-Takaoka array I*di modifikasi menjadi dua dimensi.

Adapun tahap pencarian pattern menggunakan algoritma *Zhu-Takaoka* adalah sebagai berikut :

1. *Preprocessing*
   Preprocessing dalam algoritma *Zhu-Takaoka* meliputi pencarian nilai pergeseran karakter (good-suffix shift) dan karakter tidak cocok(bad-character shift). Nilai good-suffix shift ditentukan dalam good-suffix processing sedangkan nilai bad-character shift ditentukan dalam bad-caharacter processing. Processing dilakukan sebelum proses inti dari pencarian pattern dalam suatu text.(pratama,2008)
2. *Right-to-left Scan Rule*
   Proses dalam pencarian algoritma *Zhu-Takaoka* yang dilakukan dengan teknik Right-to-left scan rule. Teknik ini yaitu denagn melakukan perbandingan antar pattern yang dicari dengan target text secara terbalik yaitu bergerak dari kanan ke kiri. Perbandingan pattern dengan target taxt dimulai dengan membandingkan karakter terakhir dari pattern (karakter paling kanan) denagn target text paling kanan. Apabila ada kecocokan maka perbandingan akan dilanjutkan denagn bergerak ke kiri sampai karakter pertama dari pattern. Sedangkan apabila terjadi ketidak cocokan maka akan dilakukan pergeseran, besarnya pergeseran yang dilakukn ditentukan oleh dua fungsi pergeseran yaitu dengan bad-character shift dan good-suffix shift.
3. *Bad-character Shift Rule*
   Aturan Bad-character shift dibutuhkan untuk menghindari pengulangan perbandingan yang tidak cocok dari suatu karakter didalam target text dengan pattern. Besarnya pergeseran yang dilakukan dalam aturan Bad-character shift disimpan dalam bentuk table array dua dimensi, tabel ini terdiri dari beberapa kolom yaitu kolom karakter dan kolom shift yang menunjukkan besarnya pergeseran yang harus dilakukan.
4. *Good Suffix Shift Rule*
   Aturan good-suffix shift dibuat untuk menangani kasus dimana terdapat pengulangan katakter pada pattern. Contoh dibawah ini akan menjelaskan bagaimana aturan bad-character shift gagal dalam menangani adanya perulangan bagian dalam pattern.

2.3 Android

Android merupakan sistem operasi bebasis Linux yang diciptakan untuk perangkat bergerak (layar sentuh) seperti handphone dan computer tablet. Android pertamakalinya dikembangkan oleh Android,Inc., dengan dukungan dari Google, yang telah membelinya pada tahun 2005. Pada tahun 2007 sistem operasi ini dipublikasikan secara resmi, bersamaan dengan berdirinya Open Handset Aliance. Ponsel Android pertamakali dipasarkan pada bulan Oktober 2008. Pengguna antarmuka Android umumnya berupa manipulasi langsung, menggunakannya dengan cara menyentuh yang serupa dengan tindakan nyata, seperti mengetik, mencubit dan menggeser untuk memanipulasi objek di layar.

2.4 Kependudukan

Kependudukan atau demografi berasal dari bahasa Yunani, demos yang berarti rakyak dan grafien yang berarti menulis. Sehingga dapat diartikan demografi adalh tulisan tulisan tentang rakyat atau penduduk. Kependudukan merupakan contoh yang berhubungan dengan struktur, alamat, agama, status perkawinan, jenis kelamin, tempat/tanggal lahir, kematian. Hingga pertahanan yang berhubungan dengan perekonomi, social, budaya serta politik. Sedangkan penduduk terdiri dari warga Negara dan orang asing yang tinggal di Negara tersebut. Jumlah penduduk dalam suatu Negara akan berubah setiap waktunya akibat dari kelahiran, kematian, perkawinan, migrasi dan mobilitas sosial.

3. ANALISA DAN PEMBAHASAN

Menjalankan prosedur preZTBc (*Preprocessing Zhu-Takaoka Bad Character*) dan preBmGs (*Preprocessing Boyer Moore Good Suffix*) untuk mendapatkan inisialisasi harus mencari tabel yaitu:

1. Mencari nilai tabel OH
2. Mencari nilai ztBc (*Zhu Takaoka Bad Caracter*)
3. Mencari nilai tabel bmGs (*Bayer Moore Good Suffic*)

Contoh algoritma zhu-takaoka
Teks: INDAH PERMATA SARI
Pattern: SARI ⟶ Panjang pattern 4 karakter

1. Mencari nilai tabel OH

Tabel 1. bmBc Awal

Index	0	1	2	3
Pattern	S	A	R	I
OH	?	?	?	?

Keterangan:
1. Index adalah index dari pattern.

ISBN: 978-602-52720-2-8
Juli 2019
Hal: 609 – 613

2. Karakter adalah pattern yang akan dicari.
3. Nilai Occurrence Heuristic (OH) adalah nilai pergeseran Bad Character.

Rumus**:**
(Panjang pattern -1 –index =)
Langkah-langkah pemberian nilainya adalah sebagai berikut:
1. Karakter pertama S index 0
   OH = 4-1-0 = 3
2. Karakter kedua A index 1
   OH = 4-1-1 = 2
3. Karakter ketiga R index 2
   OH = 4-1-2 = 1
4. Karakter keempat I index 3
   OH = 4-1-3 = 0

Jika ada karakter yang sama muncul maka diambil nilai terkecil pada OH.

Tabel 2. bmBc Akhir

Index	0	1	2	3
Pattern	S	A	R	I
OH	3	2	1	0

Tabel bmBc akhir merupakan acuan untuk menentukan tabel ztBc
Rumus**:**
(Panjang pattern -1 –index =)
Langkah-langkah pemberian nilainya adalah sebagai berikut:
5. Karakter pertama S index 0
   OH = 4-1-0 = 3
6. Karakter kedua A index 1
   OH = 4-1-1 = 2
7. Karakter ketiga R index 2
   OH = 4-1-2 = 1
8. Karakter keempat I index 3
   OH = 4-1-3 = 0

Jika ada karakter yang sama muncul maka diambil nilai terkecil pada OH.

Tabel 2. bmBc Akhir

Index	0	1	2	3
Pattern	S	A	R	I
OH	3	2	1	0

Tabel bmBc akhir merupakan acuan untuk menentukan tabel ztBc

2. Mencari nilai ztBc (*Zhu Takaoka Bad Caracter*)

Langkah selanjutnya adalah masukan semua karakter yang terdapat pada sumber teks dan buatlah menjadi dua dimensi, karakter yang dimasukkan diurutkan sesuai abjad tak berulang.

Tabel 3. ztBc Awal

	A	I	R	S	*
A					
I					
R					
S					
*					

Setelah tabel dua dimensi terbentuk, langkah selanjutnya adalah engisi tabel dengan cara melihat karakter pada baris dan karakter pada kolom. Untuk mengisi nilai pada tersebut, ada beberapa aturan sebagai berikut :

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 dituju adalah karakter pertama pada pattern, maka masukkan nilai OH karakter tersebut, tanpa melihat karakter baris.

Tabel 4. ztBc Akhir

	A	I	R	S	*
A	4	4	1	3	4

ISBN: 978-602-52720-2-8
Juli 2019
Hal: 609 – 613

I	4	4	4	3	4
R	2	4	4	3	4
S	2	0	4	3	4

3. Membuat Tabel bmGs (*Bayer Moore Good Suffix)*

Seterlah mendapatkan hasil akhir pada tabel ztBc, maka dilanjutkan dengan membuat tabel akhiran/suffix dari pattern.

Table 5. bmGs Awal

Index	0	1	2	3
Pattern	S	A	R	I
Nilai OH	3	2	1	0
Nilai MH	?	?	?	?

Keterangan:
1. Index adalah index dari pattern.
2. Karakter adalah pattern yang akan dicari.
3. Nilai Match Heuristic (MH) adalah nilai pergeseran *good suffix*, nilai MH nantinya akan digunakan ketika ditemukan kecocokan pada saat mencocokkan string karakter pertama atau lebih.

Untuk mengisi nikai MH, langkah pertama adalah membuat tabel *suffix* dari kanan ke kiri.
1. Tabel *suffix* dari kanan ke kiri, kolom *suffix* diperoleh dari pencacahan *pattern* yang dimulai dari kanan ke kiri, yaitu dari karakter A sampai karakter S.
2. tabel *suffix* dari kiri ke kanan, kolom *suffix* diperoleh dari pencacahan *pattern* yang dimulai dari kiri ke kanan, yaitu dari S sampai karakter A.

Tabel 6. *Suffix* (kanan ke kiri dan kiri ke kanan)

Pattern: SARI (Kanan Ke Kiri)				Pattern: SARI (Kanan Ke Kiri)			
Index	Prefix	Suffix	Lengh	Index	Prefix	Suffix	Lengh
3	I	SAR	3	3	S	ARI	3
2	RI	SA	2	2	SA	RI	2
1	ARI	S	1	1	SAR	1	1
0	SARI	Null	0	0		Null	0

Keterangan:
1. Pembuatan tabel *suffix* dari kanan ke kiri berfungsi untuk dijadikan *suffix comparator* pada tabel *suffix*.
2. Pembuatan *suffix* dari kiri ke kanan berfungsi sebagai *suffix* yang nantinya akan dibandingkan dan digunakan untuk mengisi nilai MH.

Tabel 7. bmGs/ztGs

Index	0	1	2	3
Pattern	S	A	R	I
Nilai OH	3	2	1	0
Nilai MH	4	4	4	1

Untuk proses *preprocessing* sudah selesai hasil dapat dilihat pada tabel 3.4 untuk tabel ztBc, dan pada tabel 3.7 untuk tabel bmGs/ztGs. Selanjutnya melakukan proses pencocokan.

Langkah 1 : Melakukan Pergeseran
Teks : INDAH PERMATA SARI
Pattern : SARI
Percobaan ke-1 :

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Teks	I	N	D	A	H		P	E	R	M	A	T	A		S	A	R	I
Pattern	S	A	R	I														

Dari hasil percobaan ke-1 karakter pattern I sejajaran dengan karakter teks A. artinya pada percobaan ini terjadi ketidakcocokan. Maka dilakukan pergeseran sejauh 4 karakter (lihat pada tabel 3.4 ztBc).

Percobaan ke-2:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Teks	I	N	D	A	H		P	E	R	M	A	T	A		S	A	R	I
Pattern					S	A	R	I										

ISBN: 978-602-52720-2-8
Juli 2019
Hal: 609 – 613

Dari hasil percobaan ke-2 karakter pattern I sejajaran dengan karakter teks E. artinya pada percobaan ini terjadi ketidakcocokan. Maka dilakukan pergeseran sejauh 4 karakter (lihat pada tabel 3.4 ztBc).

Percobaan ke-3:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Teks	I	N	D	A	H		P	E	R	M	A	T	A		S	A	R	I
Pattern									S	A	R	I						

Dari hasil percobaan ke-2 karakter pattern I sejajaran dengan karakter teks T. artinya pada percobaan ini terjadi ketidakcocokan. Maka dilakukan pergeseran sejauh 4 karakter (lihat pada tabel 3.4 ztBc).

Percobaan ke-4:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Teks	I	N	D	A	H		P	E	R	M	A	T	A		S	A	R	I
Pattern													S	A	R	I		

Dari hasil percobaan ke-2 karakter pattern I sejajaran dengan karakter teks A. artinya pada percobaan ini terjadi ketidakcocokan. Maka dilakukan pergeseran sejauh 4 karakter (lihat pada tabel 3.4 ztBc).

Percobaan ke-5:

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Teks	I	N	D	A	H		P	E	R	M	A	T	A		S	A	R	I
Pattern															S	A	R	I

Dari hasil percobaan ke-2 karakter pattern I sejajaran dengan karakter teks A. artinya pada percobaan ini terjadi ketidakcocokan. Maka dilakukan pergeseran sejauh 2 karakter (lihat pada tabel 3.4 ztBc).

4. KESIMPULAN

Dari hasil analisa dari bab-bab sebelumnya, maka dapat disimpulan yang diharapkan dapat berguna sehingga lebih bermanfaat. Adapun kesimpulan-kesimpulan tersebut, yaitu:

1. Proses pencarian data kependudukan dilakukan dengan melakukan pencocokan text dan pattern, dimana text berupa nama lengkap penduduk dan pattern kata kunci yang dicari menggunakan algoritma Zhu-Takaoka.
2. Algoritma Zhu-Takaoka dapat diterapkan dalam perancangan aplikasi pencarian data kependudukan, sehingga diharapkan memudahkan pengguna untuk mencari data kependudukan.
3. Proses pencarian yang dilakukan sangat membantu karena pencarian string dengan cepat memberikan hasil yang tepat

REFERENCES

[1] Abdi Mulia, 2018, Analisa Perbandingan Algoritma Bitap dan Algoritma Zhu-Takaoka untuk Pencarian String Pada Aplikasi Al-Quran Berbasis Mobile.
[2] Gurmanto Gatorop, Aan Erlansari, dan Funny Farady Coastera, 2017, Implementasi Algoritma Zhu-Takaoka Pada Alpikasi Kamus Istilah Musik Berbasis Android.
[3] A. K. Kusnadi, Adhi Wicaksono, "Perbandingan Algoritma Horspool Dan Algoritma Zhu-Takaoka Dalam Pencarian String Berbasis Desktop," *Ultim. Comput.*, 2017.
[4] Wikipedia, Android (Sistem Operasi) [Online], Available : https://id.m.wikipedia.org.
[5] Net, Pengertian Kependudukan dan Lingkungan Serta Hubungannya [Online], Available : http://www.pengertianku.net.
[6] Scribd, Pengertian Kependudukan [Online], Available : https://id.scribd.com.