

Aplikasi Pencarian Biografi Tokoh Politik Indonesia Berbasis Mobile Menerapkan Algoritma Colussi

Novia Intan Pratiwi, Nelly Astuti Hasibuan, Saidi Ramadan Siregar

Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: noviaintan@gmail.com

Abstrak—Negara Indonesia adalah negara demokrasi dalam menjalankan pemerintahannya. Dalam menjalankan sistem demokrasinya, masyarakat berperan penting dalam hal itu. Salah satu contohnya dalam pilkada (pemilihan kepala daerah) atau pemilu (pemilihan umum) dalam kurun waktu 5 tahun sekali. Pada saat memasuki musim 'pesta politik' tersebut, banyak terlihat foto-foto para calon politik tertempel melalui selebaran ataupun baliho-baliho di jalanan dan melakukan kunjungan-kunjungan kampanye, hal ini dilakukan para calon pemimpin dengan tujuan agar dikenal masyarakat dan dekat dengan masyarakat yang akan memilihnya. Kenyataan di lapangan, banyak masyarakat yang belum mengenal bakal calon pemimpin yang mencalonkan dirinya untuk menjadi politikus di Indonesia. Algoritma string matching yang diterapkan pada penelitian ini adalah algoritma colussi. Algoritma colussi merupakan suatu pengembangan dari algoritma knuth-morris-pratt.

Kata Kunci: Aplikasi, Pencarian Biografi Tokoh Politik, Algoritma Colussi

Abstract—The Indonesian state is a democratic country in running its government. In carrying out its democratic system, the community plays an important role in this. One example is in the pilkada (regional head elections) or elections (general elections) in a period of 5 years. At the time of entering the 'political party' season, many photos of political candidates were seen pasted through leaflets or billboards on the streets and conducting campaign visits, this was done by prospective leaders with the aim of being known to the public and close to the people who would be choose it. The reality on the ground, many people who do not know the prospective leaders who nominate themselves to become politicians in Indonesia. The string matching algorithm applied in this study is the colussi algorithm. The colussi algorithm is a development of the knuth-morris-pratt algorithm.

Keywords: Application, Biographical Search for Political Figures, Colussi Algorithm

1. PENDAHULUAN

Negara Indonesia adalah negara demokrasi dalam menjalankan pemerintahannya. Dalam menjalankan sistem demokrasinya, masyarakat berperan penting dalam hal itu. Salah satu contohnya dalam pilkada (pemilihan kepala daerah) atau pemilu (pemilihan umum) dalam kurun waktu 5 tahun sekali.

Pada saat memasuki musim 'pesta politik' tersebut, banyak terlihat foto-foto para calon politik tertempel melalui selebaran ataupun baliho-baliho di jalanan dan melakukan kunjungan-kunjungan kampanye, hal ini dilakukan para calon pemimpin dengan tujuan agar dikenal masyarakat dan dekat dengan masyarakat yang akan memilihnya. Kenyataan di lapangan, banyak masyarakat yang belum mengenal bakal calon pemimpin yang mencalonkan dirinya untuk menjadi politikus di Indonesia.

Untuk mengatasi permasalahan diatas penulis membangun aplikasi pencarian biografi tokoh politik Indonesia menggunakan bahasa pemrograman berbasis android agar dapat dioperasikan dengan mudah pada *smartphone* yang didukung sistem operasi android. Untuk mempermudah proses pencarian biografi tokoh politik Indonesia pada aplikasi pencarian biografi tokoh politik Indonesia berbasis *mobile* yang dibangun penulis menerapkan algoritma *string matching*.

Algoritma *string matching* yang diterapkan pada penelitian ini adalah algoritma *colussi*. Algoritma *colussi* merupakan suatu pengembangan dari algoritma *knuth-morris-pratt*. Algoritma *colussi* melakukan perbandingan atau pencocokan *string* dari arah kiri ke arah kanan pada teks yang terletak pada posisi yang sama dengan posisi pola dimana nilai dari fungsi kmp^{next} sedikit lebih besar dari -1 [1]

Pada penelitian sebelumnya, yaitu mengenai Aplikasi Algoritma *Colussi* Pada Pencocokan DNA dan Perbandingannya Dengan Algoritma *Knuth-Morris Pratt* disimpulkan bahwa algoritma *colussi* memiliki keunggulan dalam hal kompleksitas waktu dan kecepatan pencarian dibandingkan algoritma *Knuth-Morris Pratt* [1]. Kemudian pada penelitian mengenai Implementasi Algoritma Pecocokan *String Colussi* Pada Aplikasi Kamus Biologi Berbasis *Android* disimpulkan bahwa semakin banyak karakter yang sama antara *pattern* dan *text*, maka semakin jauh pergeseran yang dilakukan sesuai dengan nilai $h(i)$ dan $shift(i)$ pada tabel *precolussi* yang didapat sebelumnya oleh algoritma *colussi* [2]

2. METODOLOGI PENELITIAN

2.1 String Matching

String Matching merupakan salah satu algoritma yang digunakan untuk mempercepat proses pencarian kata yang diinginkan. *String matching* dibagi menjadi dua, yakni *exact matching* dan *heuristic* atau *statistical matching*. Algoritma *string matching* telah sering digunakan sebelumnya seperti contoh pada proses pencocokan *string* berdasarkan persamaan teks data yaitu *Brute Force*. Dalam hal ini, dipilih algoritma *brute force* karena algoritma ini dapat digunakan untuk melakukan pencarian *string* atau teks. Algoritma *brute force* adalah algoritma untuk mencocokkan *pattern* dengan semua teks antara 0 dan n-m untuk menemukan keberadaan *pattern* dalam teks. Berdasarkan arah pencariannya, algoritma ini diklasifikasikan sebagai algoritma yang membaca *string* dari kiri ke kanan [1].

String matching adalah proses pencarian semua kemunculan *query* yang selanjutnya disebut *pattern* ke dalam string yang lebih panjang. *String matching* dirumuskan dengan :

$$x = x [0 \dots m-1] \quad (1)$$

$$y = y [0 \dots n-1] \quad (2)$$

dimana : $x = \text{pattern}$ $m = \text{panjang pattern}$ $y = \text{Teks}$ $n = \text{Panjang teks}$ [2].

2.2 Algoritma Colussi

Algoritma Colussi merupakan suatu pengembangan dari algoritma *Knuth-Morris-Pratt*. Algoritma *Knuth-Morris-Pratt* sendiri adalah algoritma pencocokan string dengan cara memelihara informasi karakter-karakter sebelumnya untuk melakukan jumlah pergeseran yang lebih jauh. Pada algoritma *Colussi* sendiri, himpunan dari posisi pola dibagi menjadi dua sub himpunan terpisah. Lalu percobaan pencocokan berlangsung selama dua *fase*:

1. Perbandingan dilakukan dari kiri ke kanan pada teks yang terletak pada posisi yang sama dengan posisi pola dimana nilai dari fungsi *kmp"ext* sedikit lebih besar dari -1. Posisi-posisi ini dinamakan *nohole(s)*.
2. Membandingkan posisi-posisi yang tersisa (nama lainnya adalah *hole(s)*) dari arah \square kiri ke kanan.

Strategi ini memberikan dua kelebihan, yaitu:

1. Ketika terjadi ketidakcocokan pada fase pertama, maka setelah terjadi pergeseran tidak perlu untuk membandingkan teks dengan *noholes* yang dibandingkan pada percobaan sebelumnya.
2. Ketika ketidakcocokan terjadi pada fase kedua yang berarti ada *suffix* dari pola yang sama dengan dengan bagian di dalam teks, maka setelah terjadi pergeseran bagian *prefix* dari pola akan tetap sama dengan bagian dari teks tersebut, sehingga tidak diperlukan perbandingan kembali. [3].

Secara sistematis, proses yang dilakukan algoritma *Colussi* pada saat mencocokkan *string* adalah sebagai berikut

1. Pencocokan *pattern* dan teks dimulai dari karakter yang pertama.
2. Proses pencocokan akan dimulai sesuai dengan nilai pada nilai $h[i]$, jika pada nilai $h[i]$ dimulai dari angka 1 maka proses pencocokan dimulai dari kolom indeks yang ke 1 dan selanjutnya sampai nilai $h[i]$ yang terakhir.
3. Jika karakter pada teks dan *pattern* sama, maka proses pencocokan akan berlanjut pada karakter selanjutnya, namun jika karakter pada teks dan *pattern* tidak sama maka proses pencocokan string akan berpindah pada langkah pencarian selanjutnya.
4. Jika karakter pada teks dan *pattern* sama dari kiri ke kanan, dan selalu sama sampai nilai indeks yang ke $m-1$, maka algoritma *colussi* akan melakukan pencocokan kembali dari kanan ke kiri, sesuai dengan nilai $h[i]$.
5. Jika proses pencocokan menggunakan pergeseran melalui nilai $h[i]$ mencapai nilai $h[i]$ yang terakhir, maka *pattern* yang diinginkan telah didapat atau sesuai[4].

2.3 Biografi Tokoh Politik Indonesia

Biografi adalah riwayat hidup seseorang atau tokoh yang ditulis oleh orang lain. Pembaca membaca biografi orang lain biasanya ingin tahu ideologinya, kehidupannya, perjuangannya, dan lain-lain. Hal-hal tersebut yang dianggap baik tentu diteladani dan dijadikan tolak ukur dirinya [13].

Tokoh politik yaitu seseorang yang karena latar belakang sejarahnya, sepak terjangnya dalam perjuangan dan idealismenya dikenal oleh masyarakat sehingga segala pendapatnya atau pemikirannya maupun perbuatannya diikuti. Tokoh-tokoh politik biasanya mempunyai kemampuan-kemampuan karismatis, oratorik yang bersifat mempersatukan, menengahi dan dapat memanipulasikan simbol-simbol sehingga dapat mengendalikan massa [14].

Berdasarkan uraian diatas penulis menyimpulkan biografi tokoh politik indonesia adalah riwayat hidup atau latar belakang tokoh politik yang berkebangsaan indonesia yang ditulis oleh orang lain berdasarkan data yang asli sebagai penyampaian informasi mengenai data seorang tokoh politik asal negara indonesia.

3. HASIL DAN PEMBAHASAN

Indonesia merupakan negara kepulauan terbesar di dunia yang terbentang di khtulistiwa, negara Indonesia juga memiliki tokoh politik yang ingin dan bertujuan dalam melakukan perkembangan negara Indonesia melalui organisasi partai maupun independen. Hingga saat ini banyak bermunculan tokoh politik yang baru untuk bersaing di dunia politik untuk menjadi pemimpin disuatu jabatan. Tapi disamping itu banyak masyarakat yang belum mengenal tokoh politik tersebut, sehingga banyaknya masyarakat yang tidak peduli dengan jejak karir tokoh politik tersebut. Dengan adanya program ini tentu saja masyarakat dapat dengan mudah memperoleh biografi dan jejak politik tokoh tersebut, dengan begitu masyarakat dapat mempertimbangkan tokoh tersebut saat mencalonkan diri sebagai calon pemimpin atau pejabat.

Mengatasi permasalahan untuk menemukan hasil pencarian biografi tokoh politik indonesia pada penelitian penulis membangun aplikasi pencarian biografi tokoh politik indonesia berbasis *mobile* menggunakan editor *eclipse juno* dan menerapkan algoritma *string matching colussi* guna mempermudah dan meminimalisir waktu yang dibutuhkan untuk mendapatkan hasil pencarian biografi tokoh politik indonesia yang dilakukan penguana pada aplikasi pencarian biografi tokoh politik indonesia berbasis *mobile* yang dibangun pada penelitian ini.

Algoritma *string matching colussi* yang diterapkan pada aplikasi pencarian biografi tokoh politik indonesia berbasis *mobile* yang dibangun pada penelitian ini merupakan suatu pengembangan dari algoritma *string matching knuth-morris-pratt*. Algoritma *string matching colussi* pada proses pencariannya melakukan tahap-tahap perbandingan atau pencocokan

string pattern dengan *string text* dari arah kiri ke arah kanan yang terletak pada posisi yang sama dengan posisi pola dimana nilai dari fungsi *kmp"ext* sedikit lebih besar dari -1.

3.1 Penerapan Algoritma Colussi

Implementasi algoritma *colussi* merupakan tahap penyelesaian untuk melakukan pencocokan *string* dalam pencarian biografi tokoh politik Indonesia. Pencocokan *string* menggunakan algoritma *colussi* adalah seperti berikut ini :

Teks : HATTA RADJASA

Pattern : RADJASA

Tabel 1. Tabel *PreColussi*

	I	0	1	2	3	4	5	6	7
<i>x[i]</i>	R	A	D	J	A	S	A	*	
<i>hmax[i]</i>	0	1	2	3	4	5	6	7	
<i>kmin[i]</i>	0	1	2	3	4	5	6	0	
<i>rmin[i]</i>	7	0	0	0	0	0	0	0	
<i>h[i]</i>	1	2	3	4	5	6	0		
<i>shift[i]</i>	1	2	3	4	5	6	7		

Langkah-langkah pencocokan *string pattern* dengan teks :

1. Langkah pertama

<i>i</i>	0	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	H	A	T	T	A		R	A	D	J	A	S	A
<i>P</i>	R	A	D	J	A	S	A						
<i>h[i]</i>		1	2										

Pencocokan *string* dilakukan berdasarkan nilai *h[i]* yaitu dimulai dari karakter yang terdapat pada *index* 1. Terlihat pada pencocokan A dan A adalah sama maka dilanjutkan pencocokan berikutnya antara D dan T. Terlihat pencocokan D dan T tidak sama maka dilakukan pergeseran sebanyak nilai pada *Shift[1]* yaitu 2 langkah.

2. Langkah kedua

<i>i</i>	0	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	H	A	T	T	A		R	A	D	J	A	S	A
<i>P</i>			R	A	D	J	A	S	A				
<i>h[i]</i>				1									

Terlihat pada pencocokan A dan T adalah tidak sama, maka dilakukan pergeseran sebanyak nilai pada *Shift[0]* yaitu 1 langkah.

3. Langkah ketiga

<i>i</i>	0	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	H	A	T	T	A		R	A	D	J	A	S	A
<i>P</i>				R	A	D	J	A	S	A			
<i>h[i]</i>					1	2							

Terlihat pada pencocokan A dan A adalah sama, maka dilakukan pencocokan selanjutnya yaitu D dan Spasi. Terlihat pencocokan D dan Spasi adalah tidak sama maka dilakukan kembali pergeseran sebanyak nilai pada *Shift[1]* yaitu 2 langkah.

4. Langkah keempat

<i>i</i>	0	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	H	A	T	T	A		R	A	D	J	A	S	A
<i>P</i>						R	A	D	J	A	S	A	
<i>h[i]</i>							1						

Terlihat pada pencocokan A dan R adalah tidak sama, maka dilakukan pergeseran sebanyak nilai pada *Shift[0]* yaitu 1 langkah.

5. Langkah kelima

<i>i</i>	0	1	2	3	4	5	6	7	8	9	10	11	12
<i>T</i>	H	A	T	T	A		R	A	D	J	A	S	A
<i>P</i>							R	A	D	J	A	S	A
<i>h[i]</i>							0	1	2	3	4	5	6

Terlihat pencocokan *string* pada langkah kelima seluruh *pattern* memiliki kecocokan dengan teks, maka pergeseran untuk pencocokan *string pattern* dengan *string* teks diberhentikan.

3.1 Implementasi

Kebutuhan sistem dapat dilakukan untuk menganalisis suatu kebutuhan sistem yang diperlukan dalam pembangunan sistem. Kebutuhan sistem ada dua yaitu, analisis kebutuhan perangkat keras (*hardware*) dan analisis kebutuhan perangkat lunak (*software*). Analisis kebutuhan sistem dapat dijelaskan sebagai berikut :

Kebutuhan perangkat lunak (*software*) yaitu program yang diperlukan untuk melakukan proses intruksi atau menjalankan perangkat keras. Agar aplikasi dapat dibuat dan diimplementasikan sesuai perancangan, maka diperlukan suatu perangkat lunak dengan spesifikasi *software* sebagai berikut ::

a. Kebutuhan Perangkat Lunak Pengembangan

- 1) *Eclipse IDE*
- 2) *ADT (Android Development Tool)*
- 3) *SDK (Software Development Kit)*
- 4) *SQLite Database Browser*

b. Kebutuhan Perangkat Lunak Penggunaan

- 1) *Android Virtual Device Manager*

Kebutuhan perangkat keras (*Hardware*) merupakan analisa kebutuhan sistem yang digunakan untuk mengetahui secara jelas perangkat yang dibutuhkan untuk mendukung proses pengembangan dan penggunaan dari sistem aplikasi yang akan dibuat. Adapun spesifikasinya adalah sebagai berikut :

a. Kebutuhan Perangkat Keras Pengembangan :

- 1) *Procesor Intel Core i5 2.7 GHz*
- 2) *Intel HD Graphics Nvidia GT 820M*
- 3) *Memory 4GB DDR3s*
- 4) *Display 14 inch*
- 5) *Harddisk 500GB*

b. Kebutuhan Perangkat Keras Penggunaan :

- 1) *Layar 540 x 960*
- 2) *CPU Dualcore 1.4GHz*
- 3) *Memory 1.5 GB RAM*
- 4) *ROM 8GB*

Tampilan menu utama berfungsi sebagai mekanisme komunikasi antara *user* dengan aplikasi pencarian biografi tokoh politik indonesia berbasis *mobile* untuk melakukan aktivitas pemilihan menu utama yang tersedia. Tampilan menu utama pada aplikasi pencarian biografi tokoh politik indonesia berbasis *mobile* memiliki fitur menu pencarian, menu tentang, dan menu tutup. Adapun tampilan menu utama pada aplikasi pencarian biografi tokoh politik indonesia berbasis *mobile* dapat dilihat pada gambar berikut :



Gambar 1. Tampilan Menu Utama

Tampilan pencarian berfungsi sebagai mekanisme komunikasi antara *user* dengan aplikasi pencarian biografi tokoh politik indonesia berbasis *mobile* untuk melakukan pencarian biografi tokoh yang dimaksud. Tampilan pencarian pada aplikasi ini memiliki fitur *EditText* untuk menampung kata kunci yang akan dicari dan *ListView* untuk menampilkan prediksi hasil pencarian yang sesuai dengan kata kunci. Adapun tampilan pencarian pada aplikasi ini dapat dilihat pada gambar berikut :



Gambar 2. Tampilan Pencarian

Tampilan detail hasil pencarian berfungsi sebagai mekanisme komunikasi antara *user* dengan aplikasi pencarian

biografi tokoh politik indonesia berbasis *mobile* untuk menampilkan detail hasil pencarian tokoh politik yang dicari sesuai kata kunci. Tampilan detail hasil pencarian biografi tokoh politik memiliki fitur *TextView1* untuk menampilkan kata kunci yang dicari dan *TextView2* untuk menampilkan informasi dari kata kunci yang dicari. Adapun tampilan detail hasil pencarian pada aplikasi pencarian biografi tokoh politik indonesia berbasis *mobile* ini dapat dilihat pada gambar berikut :



Gambar 3. Tampilan Detail Hasil Pencarian

Tampilan tentang berfungsi sebagai mekanisme komunikasi antara *user* dengan aplikasi pencarian biografi tokoh politik indonesia berbasis *mobile* untuk menampilkan informasi tentang aplikasi yang dibangun dan juga memiliki fitur *TextView* untuk menampilkan informasi tersebut. Adapun tampilan tentang pada aplikasi ini dapat dilihat pada gambar berikut :



Gambar 4. Tampilan Tentang

4. KESIMPULAN

Mengacu kepada hasil penelitian yang telah dilakukan penulis memaparkan beberapa kesimpulan bahwa aplikasi pencarian biografi tokoh politik Indonesia berbasis *mobile* pada penelitian ini dibangun menggunakan editor *eclipse juno* dan *database SQLite*. Informasi biografi tokoh politik Indonesia dapat diperoleh dengan memanfaatkan aplikasi pencarian biografi tokoh politik Indonesia berbasis *mobile* yang menerapkan algoritma *colussi*. Aplikasi ini memudahkan *user* untuk mendapatkan informasi tentang biografi tokoh politik Indonesia sesuai dengan nama politikus yang dimasukkan pada kolom pencarian dan dirancang untuk *smartphone* android sehingga memudahkan *user* dalam penggunaannya.

REFERENCES

- [1] Raden Prana Adikhrisna, "Aplikasi Algoritma Colussi Pada Pencocokan DNA dan Perbandingannya Dengan Algoritma Knuth-Morris Pratt," Makalah IF2251 Strategi Algoritmik, 2008.
- [2] Dian Sartini, "Implementasi Algoritma Pencocokan String Colussi Pada Aplikasi Kamus Biologi Berbasis Android," Skripsi, 2016.
- [3] Mesran, "Implementasi Algoritma Brute Force Dalam Pencarian Data Katalog Buku Perpustakaan," 2014.
- [4] Fince Tinus Waruwu, "Implementasi Algoritma Knuth Morris Pratt Pada Aplikasi Kamus Istilah Latin Flora dan Fauna Berbasis Android," 2014.
- [5] Raden Prana Adikhrisna, Aplikasi Algoritma Colussi Pada Pencocokan DNA dan Perbandingannya Dengan Algoritma Knuth-Morris Pratt, Makalah IF2251 Strategi Algoritmik, 2008.
- [6] Dian sartini, Implementasi Algoritma Pencocokan String Colussi Pada Aplikasi Kamus Biologi Berbasis Android, Skripsi, Fakultas Ilmu Komputer Dan Teknologi Informasi Universitas Sumatera Utara, Medan, 2016.
- [7] Dkk. Rizal, "Rancang Bangun Aplikasi Realtime Translation Untuk Penerjemahan Bahasa Inggris Indonesia Berbasis Augmented Reality Pada Android," vol. Vol. 4, 2016.
- [8] P. P. Widodo, Menggunakan UML untuk Memodelkan Analisis & Desain Sistem Berorientasi. Bandung: Informatika, 2011.
- [9] A.S Rosa and M. Shalahuddin, Reayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Bandung: Informatika, 2014.
- [10] Nasruddin Safaat H, Pemograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android., 2015.
- [11] M. Hilmi Masruri, Java Creativity, Buku Pintar Android., 2015.

- [12] Alfa Satyaputra, Eva Maulina Aritonang, Java For Beginners with Eclipse 4.2 Juno. Jakarta: PT Elex Media Komputindo, 2012.
- [13] implementasi algoritma levenshtein distance dan algoritma knuth morris pratt pada aplikasi asmaul husna berbasis android, Jurnal Sistem Informasi, ISSN : 259 8-599X. Uli fitrianti & mutammimul ula.
- [14] Toyidin, Sastra Indonesia Puisi Prosa Drama, Pustaka Bintang, Subang, 2012.
- [15] [http://defenisikata.blogspot.com/2015/08/definisi-dan Gambar 2.2 Elipse Juno-pengertian-tokoh-politik.htm](http://defenisikata.blogspot.com/2015/08/definisi-dan-Gambar-2.2-Elipse-Juno-pengertian-tokoh-politik.htm)
- [16] A. A. B. Ginting and D. P. Utomo, "PERANCANGAN APLIKASI CATALOG WISATA DI SUMATERA UTARA MENGGUNAKAN ALGORITMA RABIN-KARP," KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer), vol. 3, no. 1, pp. 57-63, 2019.