

## Aplikasi Verifikasi Kepulangan Warga Binaan Pemasyarakatan Menggunakan Algoritma Knuth Morris Pratt

Woro Isti Rahayu <sup>1)</sup>, Ajis Trigunawan <sup>2)</sup>

Prodi/Jurusan D4 Teknik Informatika Politeknik Pos Indonesia

Jl. Sari Asih No. 54 – Bandung 40151, Indonesia Tlp +6222 2009570, Fax. +6222 200 9568

<sup>1)</sup> wistirahayu@yahoo.com, <sup>2)</sup> ajis.trigunawan45@gmail.com

<sup>2)</sup> Rumah Tahanan Klas 1 Bandung

Jl. Jakarta no.27, Kebonwaru, Kec Batununggal, Kota Bandung, Jawa Barat 40272

---

### ABSTRAK

Perkembangan teknologi saat ini yang semakin maju membuat beberapa hal yang sebelumnya dilakukan secara manual mulai dipindahkan menjadi ada campur tangan teknologi. Salah satunya adalah proses pencocokan data pada salah satu instansi negara yaitu Rumah Tahanan Negara Klas 1 Bandung khususnya bagian adm.perawatan, beberapa hal mengenai pencocokan data sangat dibutuhkan terutama perihal pencocokan data untuk kepulangan tahanan yang masih berjalan manual dan memakan waktu yang cukup lama. Untuk menyelesaikan masalah tersebut maka pembuatan aplikasi yang berisi fungsi verifikasi kepulangan tahanan untuk mempercepat pegawai yang melakukan proses kepulangan dan penerapan metode *knuth morris pratt* untuk melakukan pencocokan string pada jawaban tahanan. Maka setelah melakukan pengujian terhadap aplikasi dengan terimplementasinya metode dan aplikasi ini membuat proses verifikasi kepulangan tahanan menjadi lebih cepat dan mudah, aplikasi ini dibangun berbasis web dengan menggunakan algoritma pencocokan string *Knuth Morris Pratt*, dan dibangun dengan menggunakan bahasa pemrograman PHP dan *Framework*, analisis perancangan sistem ini menggunakan *Unified Modeling Language (UML)*.

**Kata Kunci:** Tahanan, Rumah Tahanan Negara Klas I Bandung, *Knuth Morris Pratt*.

---

### I. PENDAHULUAN

#### 1.1 Latar Belakang

Aplikasi untuk pencocokan kata yang dicari dari berbagai dokumen merupakan suatu teknologi yang sudah sering lihat dalam kehidupan sehari - hari. Aplikasi tersebut dapat digunakan untuk memudahkan pencarian suatu kata kunci yang ingin dicari dari pada melakukan pencarian secara manual dan akan memakan waktu yang cukup lama. Contoh aplikasi dalam penerapan pencarian kata seperti pencarian kata kunci pada Microsoft Word dengan menggunakan Control + F yang artinya find pencarian data pencarian kata kunci dalam suatu kamus besar, dan masih banyak lagi yang diterapkan dalam kehidupan sehari – hari[1].

Pencocokan *String* adalah suatu masalah yang hampir terjadi di seluruh bidang kehidupan, pencocokan *String* juga adalah logika untuk mencari kata / kalimat dalam pola berbasis teks. algoritma pencocokan *String* digunakan untuk memperoleh informasi secara cepat dan efisien, sebagai pencarian pada sebuah aplikasi[2]. Pencocokan *String* merupakan bagian penting dari sebuah proses pencarian *String (String searching)* dalam sebuah dokumen. Hasil dari pencarian sebuah *String* dalam dokumen tergantung dari teknik atau cara pencocokan *String* yang digunakan[3].

Salah satu teknik yang digunakan untuk membuat pencocokan *String* data yaitu menggunakan algoritma *knuth morris pratt*. Algoritma *knuth morris pratt* memecahkan

masalah dengan melakukan proses perbandingan dimulai dari kiri ke kanan [5].

Berdasarkan uraian diatas, maka peneliti membuat aplikasi verifikasi untuk keperluan wbp[6]. Aplikasi ini agar dapat memudahkan pengguna yang mencari data yang di inginkan [7], serta dapat mempercepat dalam waktu verifikasi keperluan. Pembuatan aplikasi ini akan dibuat dengan menggunakan bahasa pemrograman *PHP* dan *Database* yang digunakan yaitu *MySQL*, *sublime* serta menggunakan *framework CodeIgniter* [8]. Tahapan-tahapannya pada UML yaitu *Use Case Diagram*, *Class Diagram*, *Sequence Diagram*, *Collaboration Diagram*, *Statechart Diagram*, *Component Diagram*, *Deployment Diagram*. Dan menggunakan metode Algoritma *Knuth Morris Pratt*[9].waktu pendaftaran kunjungan yang dapat dilakukan mandiri oleh pengunjung melalui jaringan web browser.

## 1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang ada, maka dapat diidentifikasi menjadi beberapa masalah sebagai berikut:

1. Bagaimana Mengimplementasikan algoritma *knuth morris pratt*.
2. Bagaimana cara mempercepat proses verifikasi keperluan.

## 1.3 Tujuan dan Manfaat

Adapun tujuan dari penelitian ini adalah:

1. Data yang telah diinputkan akan di cocokkan oleh algoritma *knuth morris pratt* pada aplikasi tersebut dan di verifikasi.
2. Membuat aplikasi verifikasi untuk keperluan WBP menggunakan lagoritma *knuth morris pratt*.

## 1.4 Ruang lingkup

Dalam pembuatan analisis sistem ini terdapat beberapa batasan masalah yang telah penulis batasi, yaitu :

1. Pada penelitian ini mencakup tentang verifikasi data keperluan WBP.
2. Pada penelitian ini mencakup tentang keperluan WBP.

## II LANDASAN TEORI

### 2.1 String Matching

Pengolahan *String matching* atau sering disebut dengan pencocokan *String* adalah proses pencarian semua kemunculan *query* yang selanjutnya disebut *Pattern* ke dalam *String* yang lebih panjang (teks). *Pattern* yang dilambangkan dengan  $x=x[0..m-1]$  dan panjangnya adalah  $m$ . Teks dilambangkan dengan  $y=y[0..n-1]$  dan panjangnya adalah  $n$ . Kedua *String* terdiri dari sekumpulan karakter yang disebut alphabet yang di lambangkan dengan  $\Sigma$  dan mempunyai ukuran  $\sigma$ . *String matching* dibagi menjadi dua, yakni *exact matching* dan *heuristic*[10].

### 2.2 Algoritma Knuth Morris Pratt

#### 2.2.1 Penggunaan Algoritma Knuth Morris Pratt

Algoritma *Knuth Morris Pratt* pada system informasi yang akan dibangun, setelah rancangan algoritma *Knuth Morris Pratt* terbentuk maka dilakukan tahap implementasi.

Algoritma *Knuth Morris Pratt* dikembangkan oleh D. E. Knuth, bersama dengan J. H. Morris dan V. R. Pratt. Algoritma *Knuth-Morris-Pratt* merupakan pengembangan dari algoritma pencarian string sebelumnya, yaitu algoritma *Brute Force*. Algoritma *Brute-Force* merupakan algoritma dasar yang paling sederhana dalam menyelesaikan persoalan pencocokan string yang melakukan pencarian pada setiap posisi di dalam teks antara 0 dan  $n-m$ , dimana  $n$  adalah panjang teks/banyaknya nama file yang tersimpan di komputer dan  $m$  adalah panjang karakter dari suatu pattern (kata yang akan dicari) [11].

Dengan menggunakan algoritma *Knuth-Morris-Pratt*, secara rata - rata proses pencarian akan menjadi lebih cepat jika dibandingkan dengan algoritma lainnya [12].

Secara sistematis, langkah-langkah yang dilakukan algoritma *Knuth morris pratt* pada saat mencocokkan *string* yaitu :

1. Masukkan Query kata yang akan dicari. Dengan permisalan  $P = Pattern$  atau pola susunan kata yang dijadikan sebagai contoh atau pola teks yang akan dicari  $T = Teks$  atau judul dokumen.
2. Algoritma *Knuth morris pratt*mulai mencocokkan *pattern* atau pola susunan

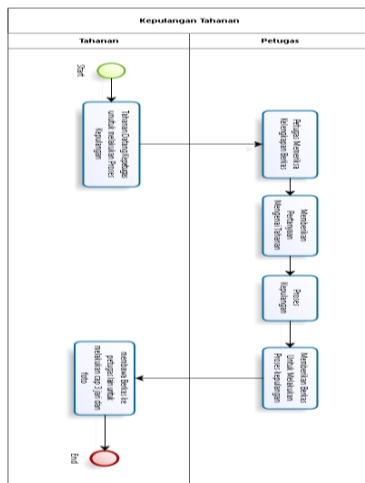
kata yang dijadikan sebagai contoh pada awal teks.

3. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* atau pola susunan kata yang dijadikan sebagai contoh dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi yaitu :
  - a. Karakter di *pattern* tau pola susunan kata yang dijadikan sebagai contoh dan di teks yang dibandingkan tidak cocok (mismatch).
  - b. Semua karakter di *pattern* atau pola susunan kata yang dijadikan sebagai contoh cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
4. Algoritma kemudian menggeser *pattern* atau pola susunan kata yang dijadikan sebagai contoh berdasarkan tabel *next*, lalu mengulangi langkah no. 2 sampai *pattern* atau pola susunan kata yang dijadikan sebagai contoh berada di ujung teks.

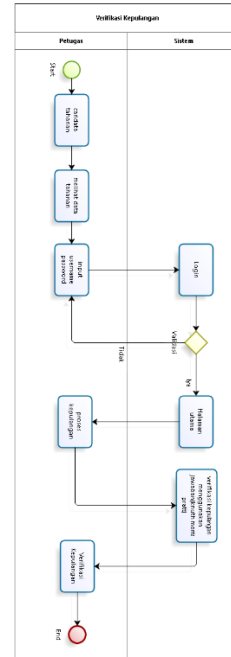
### III ANALISIS DAN PERANCANGAN

#### 3.1 Analisis

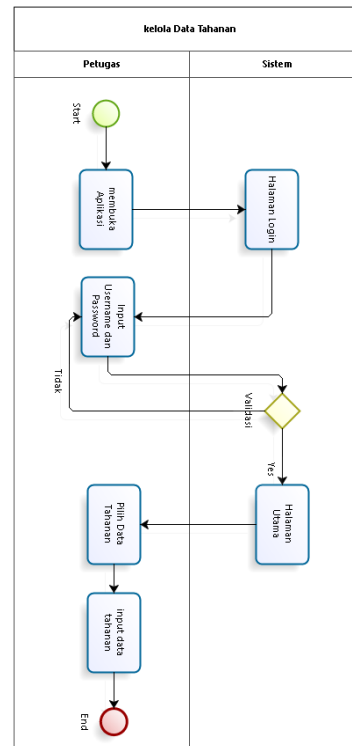
##### 3.1.1 Flow Chart Yang Sedang Berjalan



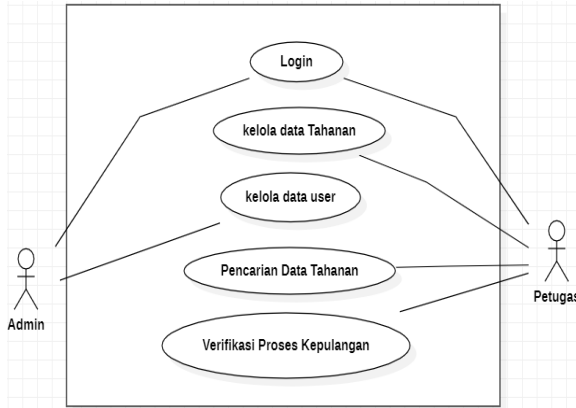
Gambar 3.1 Flow Chart Proses Kepulangan Tahanan



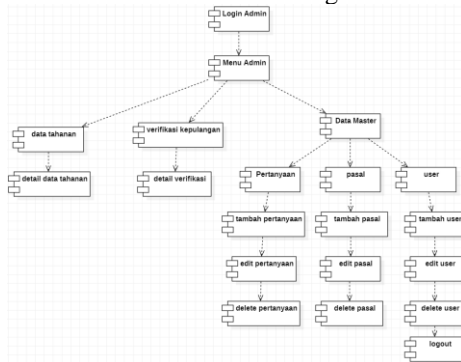
Gambar 3.2 Flow Chart Verifikasi Kepulangan



Gambar 3.3 Flow Chart Kelola Data Tahanan



Gambar 3.4 Use Case Diagram

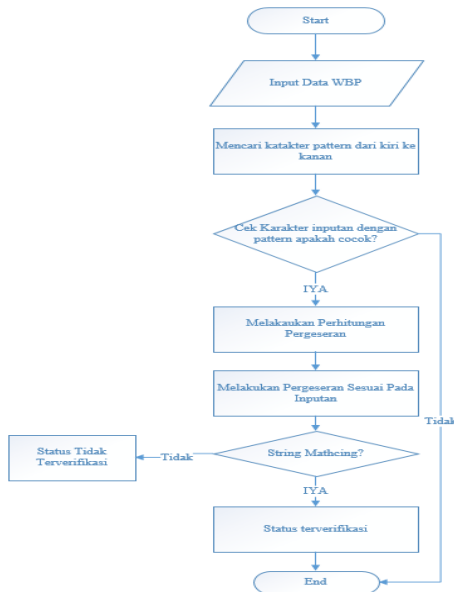


Gambar 3.5 Component Diagram

## IV PENGKAJIAN DAN EVALUASI

### 4.1 Simulasi Algoritma Knuth Morris Pratt

#### 4.1.1 Flow Chart Algoritma Knuth Morris Pratt



Gambar 3.2 diagram alur metode knuth morris pratt

### 4.2 Cara Kerja Algoritma Knuth Morris Pratt

Dalam Penelitian ini, Penulis menerapkan proses pencocokan string pada saat melakukan konfirmasi kepemilikan dengan cara pencocokan jawaban yang di mana petugas tahanan akan menkonfirmasi kepulangan tahanan dengan cara mencocokkan jawaban dari beberapa pertanyaan mengeni identitas tahanan tersebut, dan algoritma *knuth morris pratt* akan melakukan proses pencocokan jawaban tersebut, penggunaan algoritma *knuth morris pratt* Dalam melakukan pencocokan teks.

String : RISALAH

Pattern : SALAH

Tahapan pencarian pattern (P) dalam Teks (S) :

- 1 Pada pergeseran Ke-1 *Pattern* tidak cocok dengan *string*. Maka *pattern* akan bergeser satu posisi ke kanan.

Tabel 1. Pergeseran Pertama

String	R	i	s	a	l	a	h
Pattern	S	a	l	a	h		

- 2 Pada pergeseran Ke-2 *Pattern* tidak cocok dengan *string*. Maka *pattern* akan bergeser satu posisi ke kanan.

Tabel 2. Pergeseran Kedua

String	R	i	s	a	l	a	h
Pattern		S	a	l	a	h	

- 3 Pada pergeseran Ke-3 *Pattern* tidak cocok dengan *string*. Maka *pattern* akan bergeser satu posisi ke kanan.

Tabel 3. Pergeseran Ketiga

String	R	I	s	a	l	a	h
Pattern			S	a	l	a	h

- 4 Pada pergeseran Ke-4 *pattern* cocok dengan *string* Karena ada kecocokan, maka algoritma *Knuth Morris Pratt pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*.

Tabel 4. Pergeseran Keempat

- 5 Pada pergeseran Ke-5 *pattern* cocok dengan *string* Karena ada kecocokan, maka algoritma *Knuth Morris Prat pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*.

Tabel 5. Pergeseran Kelima

<i>String</i>	R	i	S	a	l	a	h
<i>Pattern</i>			S	a	l	a	h

- 6 Pada pergeseran Ke-6 *pattern* cocok dengan *string* Karena ada kecocokan, maka algoritma *Knuth Morris Prat pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*.

Tabel 6. Pergeseran Keenam

<i>String</i>	R	i	S	a	l	a	h
<i>Pattern</i>			S	a	l	a	h

- 7 Pada pergeseran Ke-7 *pattern* cocok dengan *string* Karena ada kecocokan, maka algoritma *Knuth Morris Prat pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*. Namun karena jumlah *pattern* hanya 6 huruf, maka pencarian akan dihentikan dan diperoleh hasil bahwa *pattern P* terdapat kecocokan dengan *string S* sebesar 100 persen.

Tabel 7. Pergeseran Ketujuh

<i>String</i>	R	i	s	a	l	a	h
<i>Pattern</i>			S	a	l	a	h

## V KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Setelah melakukan analisis, perancangan dan implementasi aplikasi verifikasi keputungan warga binaan pemasyarakana menggunakan framework Codelgniter, maka dapat disimpulkan sebagai beriku :

1. Algoritma Knuth Morris Pratt yang di implementasikan dapat mencocokkan data yang telah di inputkan dan hasil pencocokan tersebut akan merubah status tahanan yang akan di pulangkan.

<i>String</i>	R	I	s	a	l	a	h
<i>Pattern</i>			S	a	l	a	h

2. Dengan adanya aplikasi verifikasi untuk keputungan warga binaan pemasyarakatan menggunakan algoritma Knuth Morris Pratt akan membantu mempercepat proses verifikasi keputungan untuk warga binaan pemasyarakatan yang di mana proses ini di bantu oleh algoritma di saat pencocokan string untuk verifikasi keputungan.

## DAFTAR PUSTAKA

- [1] Z. D. P. a. E. E. Afriansyah, "Rancang Bangun Aplikasi Pencocokan Dna Manusia Menggunakan Algoritma Levenshtein Distance (Studi Kasus: DNA Kanker Hati Manusia)," *Vol. 3 No.2 November 2015*, 2015.
- [2] M. Syarif, "Implementasi Algoritma String Matching Dalam Pencarian Surah Dan Ayat Dalam Al-Quran Berbasis Web," *Vol 6, No 2 (2017)*, 2017.
- [3] E. A. J. a. S. S. Ernawati, "The Use of Documentary Video to Teach Writing News Item Text to the Tenth Grade Students in SMAN 4," *Vol 4, No 1 (20149)*, 2019.
- [4] J. N. C. M. T. a. D. K. W. Richter, "Recursive document network searching system having manual and learned component structures," 2014.
- [5] S. S. Shenoy, . U. S. Nayak and B. Neelima, "SWIFT-A Performance Accelerated Optimized String Matching Algorithm for Nvidia GPUs," 2017.
- [6] N. Sulistyowati, A. T. Widodo and W. Sumarni, "EFEKTIVITAS MODEL PEMBELAJARAN GUIDED DISCOVERY LEARNING TERHADAP KEMAMPUAN PEMECAHAN MASALAH KIMIA," *Vol 1 No 2 (2014)*, 2014.
- [7] T. Sofiansah, "SISTEM INFORMASI DONOR DARAH PALANG MERAH INDONESIA KOTA BANDUNG BERBASIS WEB," 2014.
- [8] M. Pirooznia, V. Nagarajan and Y. Deng, "GeneVenn - A web application for comparing gene lists using Venn Diagrams," 2014.
- [9] F. Boutekkouk, "Automatic SystemC Code Generation from UML Models at Early Stages of System on Chip Design," *Volume 8- No.6*, 2014.

- [10] J. I. E. B. M. a. E. B. Sinaga, "Aplikasi Mobile Pencarian Kata Pada Arti Ayat Al-Qur'an Berbasis Android Menggunakan Algoritma String Matching," *Vol. 2 No. 2 Juni 2016*, 2016.
- [11] K. W. Argakusumah and S. Hansun, "Implementasi Algoritma Boyer-Moore pada Aplikasi Kamus Kedokteran Berbasis Android," 2014.
- [12] M. F. Reza, A. Yudhana and Mushlihudin, "Monitoring Lama Waktu Parkir Motor Berbasis Web," 2017.