PENERAPAN ALGORITMA *KNUTH MORRIS PRATT* DALAM FITUR PENCARIAN PENGARSIPAN DOKUMEN PADA SMA PLUS NEGERI 17 PALEMBANG

Mohammad Ilham[1], Ahmad Haidar Mirza[2]
Fakultas Ilmu Komputer, Universitas Bina Darma
Email: mohammadilham200398@gmail.com[1], haidarmirza@binadarma.ac.id[2]

ABSTRAK

Sejalan dengan perkembangan teknologi informasi, hal itu mendorong setiap sekolah terutama bidang tata usaha untuk meningkatkan dan mengembangkan setiap informasi yang disajikan untuk menunjang pelayanan yang lebih baik. Tata usaha merupakan bagian sekolah yang memiliki wewenang menangani pengarsipan dokumen. Pengelolaan data pengarsipan dokumen di SMA Plus N 17 Palembang masih menggunakan cara manual yang akan diarsipkan dilemari arsip. Pada saat-saat tertentu terkendala dimana suatu dokumen yang sudah dibuat bertumpuk-tumpuk dengan jumlah yang banyak akan mengalami kesulit untuk ditemukan. Pemasalahan tersebut maka terdapat tiga persoalan utama dalam penelitian ini yakni pembuatan sistem fitur pencarian pengarsipan dokumen oleh admin atau staff TU dan penerapan algoritma *Knuth Morris Pratt* (KMP). Algoritma KMP adalah pencocokan *string* yang melakukan perbandingan karakter teks dan karakter pattern. Algoritma ini adalah bagaimana memanfaatkan karakter-karakter *pattern* yang sudah diketahui ada di dalam teks sampai terjadinya ketidak cocokan untuk melakukan pergeseran. Hasil yang diharapkan dari penelitian ini adalah dapat menghemat waktu dan dapat dilakukan dengan lebih baik, cepat dan mudah. Pengembangan sistem pencarian pengarsipan dokumen yang terdiri atas : XAMPP Server, Mozilla Firefox, bahasa pemrograman PHP dan HTML dan menggunakan metode *Software Development Life Cycle* (SDLC).

Kata kunci: Pencarian, Arsip, Algoritma, KMP

ABSTRACT

In line with the development of information technology, it encourages every school especially in the area of administration to improve and develop any information presented to support better services. Administration is a part of the school that has the authority to handle document archiving. Document archiving data management in SMA plus Negeri 17 Palembang still uses the manual method to storing archives. At certain times, there are problems where a document that has been created and stacked will have difficulty being found. Because of these problems, there are three main problems in this study, that is the creation of a document archive search feature system by the administrative staff and the application of the Knuth Morris Pratt (KMP) algorithm. KMP algorithm is a string matching that compares text characters and patterns character. This algorithm is how to utilize the pattern characters that are already known to exist in the text until a incompatibility occurs to make a shift. The expected result of this research is that it can save time and can do better, faster and easier to archiving. Document archive search system development consists of: XAMPP Server, Mozilla Frefox , PHP Programmer Language and HTML, and using Software Development Life Cycle (SDLC).

Keywords*: Search, Archives, Algorithm,KMP*

1. PENDAHULUAN

Setiap kegiatan atau pekerjaan didalam suatu sekolah atau organisasi diharuskan melakukan pengarsipan dokumen yang menjadi informasi bermanfaat bagi sekolah atau organisasi tersebut, baik digunakan sebagai bukti maupun sebagai bahan dalam proses pengambilan suatu keputusan. Algoritma *Knuth Morris Pratt* merupakan salah satu pencocokan *string matching*. Metode pencarian pada KMP ini bekerja dengan melewatkan pencocokan *pattern* dengan teks yang tidak berguna untuk menghindari besarnya jumlah pencocokan. Gagasan utama dari algoritma KMP adalah pencocokan pola akan berjalan secara efisien bila kita mempunyai tabel yang menentukan berapa panjang kita seharunya menggeser seandainya terdeteksi ketidakcocokan di suatu karakter dari pola sehingga ketika terjadi ketidakcocokan di satu lokasi, kita segera tahu jumlah maksimum untuk menggese pola yang didapat sebelum melakukan pencarian. Hal ini akan menghemat perbandingan, yang selanjutnya akan meningkatkan kecepatan pada proses pencarian data [2].

SMA Plus Negeri 17 Palembang memiliki bidang tata usaha yang salah satu tugasnya adalah mengelola data/dokumen penting seperti pengarsipan dokumen berupa buku agenda (surat masuk, surat keluar dan surat internal) dari beberapa kategori tersebut disusun dan diletakan pada rak arsip yang tersedia didalam ruangan tata usaha. Staff tata usaha sering mengalami kesulitan dalam mencari dokumen dengan membuka tumpukan dokumen yang ada didalam rak arsip. dokumen yang sedikit belum menjadi masalah, namun setelah dokumen yang sudah dibuat bertumpuk-tumpuk bahkan sampai bertahun-tahun dengan jumlah yang banyak. maka hal ini menyebabkan proses pencarian dokumen menjadi lambat bahkan berkemungkinan dokumen tersebut tidak ditemukan.

Berdasarkan kondisi tersebut untuk meningkatkan pelayanannya maka diperlukan penerapan algoritma *Knuth Morris Pratt* pada pencarian pengarsipan dokumen ke dalam sebuah basis data yang harus ditunjang dengan efisiensi proses pencarian pengarsipan dokumen yang tersimpan tersebut dan dapat memberi kemudahan pada staff TU dalam melakukan pencarian data yang dilakukan secara *real time* serta dapat dengan cepat mengetahui letak kata yang dicari.

2. METODOLOGI PENELITIAN

Dalam penelitian ini metode yang digunakan adalah metode deskriptif. Metode deskriptif adalah suatu metode dalam meneliti suatu objek yang dapat mengemukakan masalah dengan mengumpulkan data-data yang bertujuan untuk mendapatkan gambaran yang jelas mengenai suatu keadaan dengan cara menyajikan, mengumpulkan dan menganalisis data tersebut sehingga menjadi informasi yang dapat digunakan untuk menganalisa dan mengambil kesimpulan mengenai masalah yang sedang diteliti.

Algoritma Knuth Morris Pratt

Algoritma KMP adalah salah satu algoritma pencarian *string,* dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, namun keduanya mempublikasikannya secara bersamaan pada tahun 1977 [1]. Jika kita melihat algoritma *brute force* lebih mendalam, kita mengetahui bahwa dengan mengingat beberapa perbandingan yang dilakukan sebelumnya kita dapat meningkatkan besar pergeseran yang dilakukan. Hal ini menghemat perbandingan, yang selanjutnya akan meningkatkan kecepatan pencarian.

Secara Sistematis, langkah-langkah yang dilakukan algoritma KMP pada saat pencocokan *string* adalah sebagai berikut [3]:

1) Algoritma KMP mulai mencocokan *pattern* pada awal teks.
2) Dari kini ke kanan, algoritma ini akan mencocokan karakter per karakter *pattern,* dengan karakter di teks yang bersesuaian sampai salah satu kondisi berikut terpenuhi

a. Karakter di *pattern* dan diteks yang dibandingkan tidak cocok *(mismatch)*
b. Semua Karakter di *pattern* cocok. Kemudian algoritma akan memberitahu penemuan posisi ini.

3) Algoritma kemudian menggeser pattern berdasarkan table next, lalu menghitung langkah 2 sampai pattern berada diujung teks.

Algoritma Knuth-Morris-Pratt (KMP) merupakan salah satu algoritma pencocokan pola. Metode pencarian KMP bekerja dengan melewatkan perbandingan-perbandingan yang tidak diperlukan untuk menghindari besarnya jumlah perbandingan, dengan demikian mencapai waktu berjalan O(n+m) yang optimal dalam kasus terburuk (*worst case*) pencocokan pola algoritma harus memeriksa semua karakter teks dan semua karakter dari pola setidaknya sekali. Ide utama algoritma KMP adalah untuk preprocess string pola *P* sehingga untuk menghitung fungsi kegagalan *f* yang menunjukan pergesaran *P* yang tepat sehingga kita dapat menggunakan kembali perbandingan yang dilakukan sebelumnya [2].

1) Menghitung Fungsi Kegagalan *f*

Proses awal algoritma KMP pola P adalah menghitung fungsi kegagalan *f*. Fungsi kegagalan *f(j)* didefinisikan sebagai panjang awalan *P* terpanjang yang merupakan akhiran dari *P[1….j]*. Algoritma Kegagalan KMP adalah:

```
1. Input : String P (pattern) dengan karakter m
2. Output: fungsi kegagalan f untuk P, panjang awalan terpanjang pattern yang merupakan
   akhiran pattern dari P[1..j]
3. i <- 1
4. j <- 0
5. f(0) <- 0
6. while i < m do
7.       if P[j] = P[i] then
8.          f(i) <- j+1
9.          i <- i+1
10.         j <- j+1
11.      else if j>0 then
12.         j <-f(j-1)
13.      else
14.         f(i) <- 0
15.         i <- i+1
16. endwhile
```

Fungsi kegagalan KMP *f(j)* untuk string pola *P* = "abacab" adalah seperti ditunjukan pada tabel berikut.

Tabel 1. Fungsi Kegagalan KMP

j	0	1	2	3	4	5
P[j]	a	b	a	c	a	b
f(j)	0	0	1	0	1	2

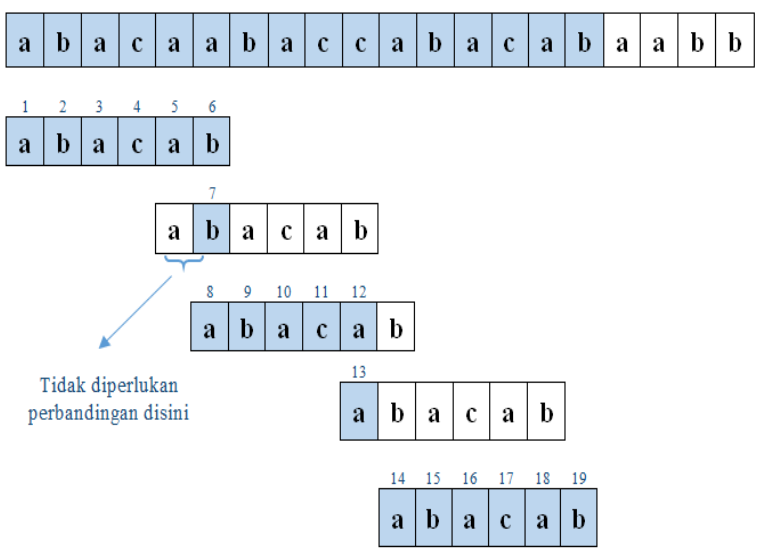
2) Pencocokan Pola KMP

Algoritma pencocokan pola KMP secara bertahap memproses string teks*T* membandingkannya dengan string pola *P*. Setiap kali ada kecocokan indeks akan bertambah namun jika ditemukan

ketidakcocokan dan sebelumnya telah membuat kemajuan dalam *P*, maka fungsi kegagalan akan menentukan indeks baru di *P* di mana harus terus memeriksa *P* terhadap *T*. Jika ditemukan ketidakcocokan dan berada di awal *P*, cukup menambah indeks untuk *T* (dan menjaga variabel indeks untuk *P* di awal). Proses akan diulangi sampai ditemukan kecocokan *P* dalam *T* atau indeks untuk *T* mencapai *n*, panjang *T* (menunjukkan bahwa tidak ditemukan pola *P* dalam *T*).
Algoritma Pencocokan Pola KMP (T,P):

```
1. Input : String T (text) dengan karakter n dan String P (pattern) dengan karakter m
2. Output: index awal dari substring pertama T cocok dengan P atau indikasi bahwa P bukan
   substing  T.
3. F <- FungsiKegagalanKMP(P)
4. i <- 0
5. j <- 0
6. while i < n do
7.       if P[j] = T[i] then
8.          if j = m-1 then
9.                return i-m+1
10.         i <- i+1
11.         j <- j+1
12.      else if j > 0 then
13.         j <- f(j-1)
14.      else
15.      i <- i+1
16. return “tidak ada substing dari t cocok dengan p”
```

Berikut contoh eksekusi dari algoritma pencococokan pola KMP pada string teks *T*=”abacaabaccabacabaabb” dan string input pola *P*= “abacab”. Fungsi kegagalan digunakan untuk menghindari mengulangi salah satu perbandingan antara string pola (*P*) dan string teks (*T*).



Gambar 1. Contoh Eksekusi Algoritma Pencocokan Pola KMP

Jika *k* adalah jumlah total dimana pola *P* telah bergeser sehubungan dengan teks *T*. Perhatikan bahwa selama eksekusi algoritma, kita memiliki $k \leq n$. Salah satu dari tiga kasus berikut terjadi pada setiap iterasi loop

a. Jika *T[i] = P[j]*, maka *i* bertambah 1, dan *k* tidak berubah, karena *j* juga bertambah 1.
b. Jika $T[i] \neq P[j]$ *and* $j>0$, maka *i* tidak berubah dan *k* bertambah setidaknya 1, karena, dalam hal ini, *k* berubah dari *i-j to i - f(j -1)*, yang merupakan tambahan dari *j - f(j–1)*, yang positif karena $f(j–1) < j$.
c. Jika $T[i] \neq P[j]$ *and* $j \neq 0$, maka *i* bertambah 1 dan *k* bertambah 1, karena *j* tidak berubah.

Kelebihan dan kekurangan Algoritma Knuth Moriss Pratt

1. Kelebihan Algoritma KMP
   Salah satu kelebihan algoritma KMP ketimbang algoritma brute force adalah adanya otak buatan yang memungkinkan tidak terjadinya pencocokan yang tidak berguna sehingga pencocokan menjadi semakin kecil
2. Kekurangan Algoritma KMP
   Kurang signifikan dalam memberikan hasil yang optimal ketika jumlah alphabet semakin banyak karena adanya kemungkinan kesalahan pencocokan.

3. HASIL DAN PEMBAHASAN

3.1 Penerapan Algoritma *Knuth Morris Pratt*

Langkah awal dalam penerapan algoritma KMP adalah perhitungan fungsi pinggiran dari sebuah patter (P) dapat disebut processing (proses awal). Tujuan dari processing ini adalah menentukan seberapa banyak loncatan atau pergeseran yang akan dilakukan ketika pencocokan *pattern* dengan teks yang tidak diperlukan.

1) Tentukan terlebih dahulu *pattern* (P) dan *text* (T). Dalam studi kasus ini *pattern* dan *text* yang ingin dicari adalah sebagai berikut.
   P = Yeyen
   T = Eka Yesi Yeyen

2) Perhitungan fungsi pinggiran
   Keterangan :
   J = Panjang karakter yang diberikan *pattern*
   P(J) = Karakter ke-j pada *pattern*
   F(J) = Nilai fungsi pinggiran dari *pattern*
   a. F(0) = 0 (karena awal dari fungsi pinggiran ditetepkan dengan nilai 0)

J	0	1	2	3	4
P(J)	Y	E	Y	E	N
F(J)	0				

   b. F(1) = pencocokan P(0) dengan Pnext(1)
   P(0) = Y dan P(1) = E
   P(0)=P(1) → Tidak terjadinya kecocokan
   Maka, F(1)= 0

J	0	1	2	3	4

P(J)	Y	E	Y	E	N
F(J)	0	0			

c. F(2) = pencocokan P(0) dengan Pnext(2)
P(0) = Y dan P(2) = Y
P(0)=P(2) → Terjadinya kecocokan(sama)
Maka, F(2)= 1

J	0	1	2	3	4
P(J)	Y	E	Y	E	N
F(J)	0	0	1		

d. F(3) = pencocokan P(1) dengan Pnext(3)
P(1) = E dan P(3) = E
P(1)=P(3) → Terjadinya kecocokan(sama)
Maka, F(3)= 2

J	0	1	2	3	4
P(J)	Y	E	Y	E	N
F(J)	0	0	1	2	

e. F(4) = pencocokan P(2) dengan Pnext(4)
P(2) = Y dan P(4) = N
P(2)=P(4) → Tidak terjadinya kecocokan
Maka, F(4)= 0

J	0	1	2	3	4
P(J)	Y	E	Y	E	N
F(J)	0	0	1	2	0

Setelah mendapatkan nilai fungsi pinggiran dari masing-masing karakter *pattern* maka langkah selanjutnya adalah penerapan algoritma KMP dengan mencocokan string *pattern* dengan *text* yang dimulai pada awal *teks.* Dilakukan dari kiri ke kanan sampai dengan semua karakter *pattern* cocok.
Keterangan :
I = Index
T = Teks
P = Pattern

Langkah pertama pencocokan awal yaitu dari kiri teks dan kiri pattern. Pada T[0] = “E” dan P[0] = “Y”. Tidak terjadinya kecocokan maka dari itu akan dilakukan pergeseran satu kali

I	0	1	2	3	4	5	6	7	8	9	10	11
T	e	k	a	y	e	s	i	y	e	y	e	n

P	y	e	y	e	n							

Langkah kedua akan dilakukan pencocokan kembali yaitu pada T[1] = “K” dengan P[1] = “Y” . Tidak terjadinyan kecocokan maka akan dilakukan pergeseran sebanyak satu kali saja.

I	0	1	2	3	4	5	6	7	8	9	10	11
T	e	k	a	y	e	s	i	y	e	y	e	n
P		y	e	y	e	n						

Langkah ketiga akan dilakukan pencocokan kembali yaitu pada T[2] = “A” dengan P[2] = “Y” . Tidak terjadinyan kecocokan maka akan dilakukan pergeseran sebanyak satu kali saja.

I	0	1	2	3	4	5	6	7	8	9	10	11
T	e	k	a	y	e	s	i	y	e	y	e	n
P			y	e	y	e	n					

Langkah keempat akan dilakukan pencocokan kembali yaitu pada T[3] = “Y” dengan P[3] = “Y”. Terjadinya kecocokan. Maka dari itu akan dilakukan pencocokan selanjutnya dengan tidak melakukan pegeseran.

I	0	1	2	3	4	5	6	7	8	9	10	11
T	e	k	a	y	e	s	i	y	e	y	e	n
P				y	e	y	e	n				

Langkah kelima akan dilakukan pencocokan lagi yaitu pada T[4] = “E” dengan P[4] = “E”. Terjadinya kecocokan. Maka tidak dilakukan pergeseran dan akan dilakukan pencocokan kembali.

I	0	1	2	3	4	5	6	7	8	9	10	11
T	e	k	a	y	e	s	i	y	e	y	e	n
P				y	e	y	e	n				

Langkah keenam akan dilakukan pencocokan yaitu pada T[5]= “S” dan P[5]= “Y”. Tidak terjadinya kecocokan maka dari itu akan dilakukan pergeseran lebih jauh dengan memanfaatkan informasi yang disimpan. Rumus pergeseran adalah i – b, i merupakan index *pattern* yang cocok dengan *teks* yakni sebesar 2 karakter dan b adalah nilai fungsi pinggiran yang indexnya terpanjang yakni 0. Maka 2-0 = 2. sehingga *pattern* digeser sebanyak 2 kali ke kanan.

I	0	1	2	3	4	5	6	7	8	9	10	11
T	e	k	a	y	e	s	i	y	e	y	e	n
P				y	e	y	e	n				

Langkah ketujuh akan dilakukan pencocokan yaitu pada T[5] = “S” dengan P[5]= “Y”. Tidak terjadinya kecocokan maka akan dilakukan pergeseran satu kali saja.

I	0	1	2	3	4	5	6	7	8	9	10	11
T	e	k	a	y	e	s	i	y	e	y	e	n
P						y	e	y	e	n		

Langkah kedeleapan akan dilakukan pencocokan yaitu pada T[6] = “I” dengan P[6]= “Y”. Tidak terjadinya kecocokan maka akan dilakukan pergeseran satu kali saja.

I	0	1	2	3	4	5	6	7	8	9	10	11
T	e	k	a	y	e	s	i	y	e	y	e	n
P							y	e	y	e	n	

Langkah kesebelas akan dilakukan pencocokan selanjutnya yaitu T(7) sampai dengan T(11)= “YEYEN” dengan P(7) sampai dengan P(11) = “YEYEN” karena ditemukan pola keseluruhan *pattern* semua karakternya sudah cocok, sehingga proses pencocokan string *pattern* dengan *teks* selesai.

I	0	1	2	3	4	5	6	7	8	9	10	11
T	e	k	a	y	e	s	i	y	e	y	e	n
P								y	e	y	e	n

3.2 Tampilan Antarmuka Pencarian Arsip Dokumen Menggunakan Algoritma KMP

Gambar 2. Tampilan Antarmuka Pencarian Arsip Dokumen Berdasarkan Surat Keluar

Antar muka pencarian arsip dokumen berdasarkan kategori surat keluar dan alamat dengan pencarian “Kepsek” ditemukan hasil pencarian sebanyak 1 dokumen dalam posisi string ke-10 dengan waktu 0.001 detik. Pada kata “Kepsek” didalam alamat terdapat bold berwarna merah yang menandakan posisi kata yang ditemukan oleh KMP. Adapun antar muka pencarian arsip dokumen berdasarkan kategori surat masuk dan perihal dapat dilihat pada gambar 2

Gambar 3. Tampilan Antarmuka Pencarian Arsip Dokumen Berdasarkan Surat Masuk

Antar muka pencarian arsip dokumen berdasarkan kategori surat masuk dan alamat pengirim dengan pencarian "Pertamina" ditemukan hasil pencarian sebanyak 1 dokumen dalam posisi string ke-12 dengan waktu 0.001 detik. Pada kata "Pertamina" didalam alamat pengirim terdapat bold berwarna merah yang menandakan posisi kata yang ditemukan oleh KMP. Adapun antar muka pencarian arsip dokumen berdasarkan kategori surat masuk dan alamat pengirim dapat dilihat pada gambar 3

4. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian sistem yang telah dilakukan maka dapat ditarik kesimpulan sebagai berikut :

1) Algoritma *Knuth-Morris-Pratt* berhasil diterapkan pada pencarian pengarsipan dokumen di SMA Plus N 17 Palembang. Hal ini memudahkan staff TU dalam mencari arsip dokumen dengan cepat dan menemukan letak kata didalam kalimat berdasarkan kategori yang dipilih. Sehingga menghasilkan pencarian arsip dokumen secara tepat dan akurat
2) Hasil pengujian performa menunjukkan bahwa rata-rata performa algoritma *Knuth-Morris-Pratt* dapat ditemukan pada form pencarian arsip dokumen sebesar 0.0017detik dengan dokumen sebanyak 100 data dokumen. Hal ini menunjukkan bahwa Algoritma *Knuth-Morris-Pratt* sudah cukup cepat dan optimal dalam menemukan kata yang dicari didalam sebuah kalimat.

DAFTAR PUSTAKA

[1] Astuti, W. (2017). *Analisis String Matching pada Judul Skripsi Dengan Algoritma Knuth-Morris-Pratt (KMP). Jurnal Ilmiah*, 167-172.

[2] Goodrich, M. T., Tamassia, R., & Mount, D. (2011). *Data Structure & Algorithms in C++*. United States Of America : John Wiley & Sons, Inc.

[3] Nursobah, & Pahrudin, P. (2019). *Penerapan Algoritma Pencarian Knuth Morris Pratt (KMP) Dalam Sistem Informasi Perpustakaan SMK TI Pratama*. Sebatik, 112-115.