

- ✍ **Sistem Informasi Piutang Berbasis Web (Halaman : 242 - 251)**
Arfan Yunus
- ✍ **Pengembangan Sistem Pakar Untuk Diagnosa Penyakit Leochorrore Pada Wanita (Halaman : 252 - 267)**
Ahmad M. Mile
- ✍ **Analisis Dan Pengembangan Jaringan Komputer Balai Pengkajian Dan Pengembangan Informasi Wilayah VII Makassar (Halaman : 268 - 286)**
Herlinah
- ✍ **Simulasi Sistem Keamanan Perparkiran Berbasis Komputer (Halaman : 287 - 296)**
Hazriani
- ✍ **Algoritma String Matching Dengan Menggunakan Modifikasi Algoritma Tuned Boyer Moore. (Halaman : 297 - 306)**
Najariah Umar
- ✍ **Simulasi Sistem Pengendali Palang Penghalang Parkiran Berbasis Lan (Local Area Network). (Halaman : 307 - 319)**
Dimiyati
- ✍ **Guru Masa Depan, Guru Yang Berteknologi (Halaman : 320 - 325)**
Syafril Ariel
- ✍ **Perancangan Jaringan Hotspot Pendukung Nomadic (Halaman : 326 - 338)**
Sarif Arbi A. Mappa

KHARISMA.TECH

Jurnal Ilmu Komputer

ISSN : 1907 - 2317

Terbit dua kali dalam setahun beserta nomor suplemen. Berisi tulisan yang diangkat dari hasil penelitian atau kajian analitis-kritis di bidang informatika dan teknologi informasi dan atau pembelajarannya.

Pelindung

Ketua STMIK KHARISMA Makassar

Ketua Penyunting

Ir. Moh. Sofyan S. Thayf

Sekretaris Penyunting

Restiana, S. T.

Penyunting Pelaksana

Sudirman, S. Kom.

Agus Halid, S. Kom.

Dimyati, S. Kom.

Arfan Yunus, S. E.

Drs. Abdullah

Penyunting Ahli

Drs. Suarga, M. Sc., M. Math., Ph. D. (Universitas Hasanuddin Makassar)

Drs. Abd. Rahman Mallala, M. Pd. (Universitas Negeri Makassar)

Drs. Abdul Halik (STIE Indonesia Makassar)

Syaiful Rahman, S. Kom., M. Kom. (STMIK KHARISMA Makassar)

M. Syukri Mustafa, S. Si., M. MSI. (STMIK Dipanegara Makassar)

Pelaksana Tata Usaha

Hamdan Arfandy, S. Kom.

Adi Mulya, S. Kom.

Alamat Penyunting dan Tata Usaha: Kampus STMIK KHARISMA Makassar, Jl Baji Ateka No. 20, Makassar. No. Telpon 0411-871555, Fax 0411-870098. Email : tech@kharisma.ac.id.
Penyunting menerima sumbangan tulisan yang belum pernah diterbitkan dalam media lain. Naskah diketik di atas kertas HVS A4 spasi 1 ½ 10-15 halaman, dengan format sebagaimana keseragaman format, istilah, dan gaya selingkung **KHARISMA.TECH**.
Harga langganan Rp 25.000,- (Dua puluh lima ribu rupiah) per eksemplar (dua kali terbit), tidak termasuk ongkos kirim.

ALGORITMA STRING MATCHING DENGAN MENGGUNAKAN MODIFIKASI ALGORITMA TUNED BOYER MOORE.

Oleh :
Najirah Umar

Abstrak : Algoritma *String matching* adalah algoritma untuk mencari kata pada kalimat. Kata disebut sebagai Pola (*Pattern*), sedangkan kalimat disebut sebagai teks (*text*). Hasil pencarian pola pada teks memiliki dua kemungkinan, yaitu pola yang dicari tidak ditemukan pada teks, atau pola yang dicari ditemukan pada posisi tertentu di dalam teks.

Algoritma *String matching*, diantaranya adalah Algoritma Tuned Boyer Moore dan Algoritma Quick Search yang keduanya memakai teknik yang sama yaitu teknik pelurusan. Tetapi pada Algoritma Tuned Boyer Moore terdapat pola yang statis sehingga algoritma ini kurang efisien karena waktu yang digunakan untuk mendapatkan karakter teks yang sama dengan pola lebih banyak. Untuk meminimalkan pola yang statis tersebut diadopsi teknik penentuan patokan pola dari Algoritma Quick Search yang menghasilkan algoritma baru yang dinamakan Algoritma AFI. Pada Algoritma AFI metode yang digunakan untuk mengukur efisiensi waktu kerja yaitu dengan cara menghitung berapa jumlah pencocokan antara karakter teks dengan karakter pola yang terjadi, dan berapa jumlah pergeseran pola yang terjadi.

Dari hasil implementasi dan pengujian diperoleh hasil dimana untuk panjang teks yang semakin besar menunjukkan semakin efisiennya Algoritma AFI dibandingkan Algoritma Tuned Boyer Moore.

Kata Kunci : Algoritma *String matching*, Algoritma Quick Search, Algoritma Tuned Boyer Moore

A. Pendahuluan

String Matching algorithm adalah algoritma untuk pencarian string pada dokumen yang umumnya terdapat pada program aplikasi yang berorientasi pada pengolahan kata atau *text editor*. String yang dicari biasa juga disebut sebagai pola (*pattern*), dan dokumen yang tersedia disebut sebagai teks (*text*). Program aplikasi *string matching* mengandung mekanisme untuk melakukan pencarian langsung string tertentu pada dokumen yang tersedia. Pencarian string pada dokumen bertujuan diantaranya untuk memodifikasi string atau menghapus string yang salah, karena mungkin salah ketik atau salah kosa kata, juga bertujuan untuk sekedar mencari informasi.

Untuk memodifikasi Algoritma Tuned Boyer-Moore yaitu dengan mengadopsi cara kerja dari salah satu Algoritma *String Matching* lainnya yaitu Algoritma Quick Search, dimana cara kerja yang diadopsi yaitu cara menentukan karakter teks sebagai patokan pergeseran pola, yang bertujuan untuk meminimalkan jumlah pola yang statis.

B. Algoritma Tuned Boyer-Moore

Algoritma Tuned Boyer-Moore (Algoritma TBM) (Charras, dkk,1997) adalah salah satu Algoritma *String Matching* yang dikembangkan dari Algoritma Boyer Moore yang mudah dipahami dan mudah diaplikasikan dalam program berbasis pengolah kata atau teks editor. Algoritma Tuned Boyer-Moore memiliki prinsip kerja yang sama dengan Algoritma Quick Search dan Algoritma Skip Search yaitu teknik pelurusan. Algoritma Tuned Boyer-Moore menggunakan tabel *tbm_bc* singkatan dari Tuned Boyer-Moore *bad character* untuk menentukan panjang pergeseran pola.

Teknik pelurusan ini akan memilih karakter pada teks sebagai patokan pergeseran pola. Pola akan digeser dengan cara meluruskan karakter dari pola yang sama dengan karakter dari Patokan Pergeseran Pola. Gambar 1 menunjukkan sebuah contoh penggunaan teknik pelurusan pola dari Algoritma Tuned Boyer-Moore. Pada proses pencocokan pola 'G O L' pada 3 karakter pertama dari teks yaitu 'A L G', pencocokan pertama kali dilakukan pada karakter terakhir dari pola yaitu 'L' dengan karakter pada teks yang sejajar dengan 'L' yaitu 'G'. Pada pencocokan pertama ini ditemukan ketidaksesuaian antara karakter pola dan karakter teks yang sejajar. Kemudian pola digeser ke kanan dimana karakter pada posisi ke-2 dari teks yaitu 'O' sebagai patokan pergeseran pola. Pola digeser dengan cara meluruskan antara karakter 'O' dari pola dengan karakter dari patokan pergeseran pola yaitu karakter 'G'.

Posisi :	0	1	2	3	4	5	6	7	8
A		L		<u>G</u>					
G		O		L					
				<u>G</u>					

Teks :
 Pola :
 Pola :

Gambar 1
 Contoh penggunaan teknik pelurusan dari Algoritma Tuned Boyer Moore

Algoritma Tuned Boyer-Moore memerlukan tabel t_{bm_bc} (Tuned Boyer Moore bad character) untuk menentukan posisi karakter-karakter dalam pola, seperti contoh diatas.

Algoritma Tuned Boyer Moore

Input : Sebuah string T dengan panjang n, dan sebuah string P dengan panjang m

Output : Posisi ditemukannya P pada T ditampilkan ke layar.

Procedure : TBM (T,P : string; n,m : integer)

1. (Proses inisialisasi array $t_{bm_bc}(0)$ sampai dengan $t_{bm_bc}(255)$ dengan nilai m)
 - For** i = 0 To 255 **Do**
 - 1.1 (Isi array t_{bm_bc} pada indeks i dengan nilai m) $t_{bm_bc}(i) \leftarrow m$
2. (Perubahan isi array t_{bm_bc} sesuai dengan letak karakter dalam string P)
 - For** i = 1 To m-1 **Do**
 - 2.1 (ganti isi array t_{bm_bc} pada indeks $ord(P(i))$ dengan nilai m-i)
 - $t_{bm_bc}(ord(p(i))) \leftarrow m-i$
3. (Banyaknya pergeseran pola untuk karakter ke-m dari pola dialihkan ke variabel shift, kemudian $t_{bm_bc}(ord(P(m)))$ diisi dengan nilai 0)
 - 3.1 $shift \leftarrow t_{bm_bc}(ord(P(m)))$;
 - 3.2 $t_{bm_bc}(ord(P(m))) \leftarrow 0$;
4. (Pertama kali j menunjukkan posisi awal pola pada teks)
 - $j \leftarrow 1$
5. (Lakukan pergeseran dan pencocokan pola pada teks selama $j \leq n-m+1$)
 - While** $j \leq n-m+1$ **Do**
 - (Pola digeser beberapa kali lalu karakter teks dengan karakter terakhir dari pola yang seletak dibandingkan)
 - 5.1 $f \leftarrow t_{bm_bc}(T(j+m-1))$;
 - 5.2 **While** ($f <> 0$) **Do**
 - 5.2.1 $j \leftarrow j + f$; $f \leftarrow t_{bm_bc}(T(j + m - 1))$;
 - 5.2.2 $j \leftarrow j + f$; $f \leftarrow t_{bm_bc}(T(j + m - 1))$;
 - 5.2.3 $j \leftarrow j + f$; $f \leftarrow t_{bm_bc}(T(j + m - 1))$;
 - (Lakukan pencocokan antara pola $P(1, \dots, m-1)$ dengan teks $T(j, \dots, j+m-1)$)
 - 5.3 $i \leftarrow 1$
 - 5.4 **While** ($i < m$) and ($T(j+i-1) = P(i)$) **Do**
 - 5.4.1 $i \leftarrow i + 1$

(Periksa apakah pola ditemukan pada teks)

5.5 If ($i = m$) then

5.5.1 Write (j)

(Geser posisi awal pola sebanyak nilai shift)

5.6 $j \leftarrow j + \text{shift}$

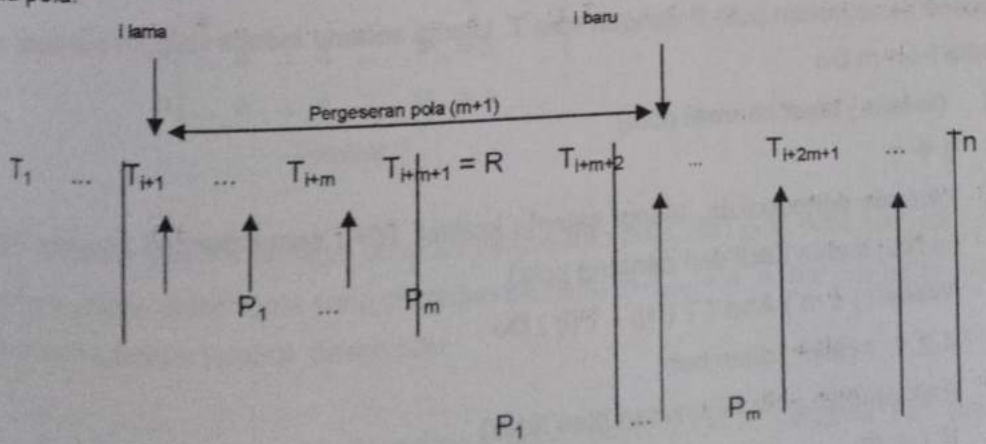
6. (Selesai)

Return

C. Algoritma Quick Search

Pada Algoritma Quick Search (QS), proses pencocokan pola $P_1 \dots P_m$ dengan teks $T_{i+1} \dots T_{i+m}$ dilakukan dari kiri ke kanan. Pencocokan pertama kali dilakukan antara karakter pertama dari $P_1 \dots P_m$ dengan karakter pertama dari $T_{i+1} \dots T_{i+m}$, apabila sesuai maka pencocokan dilanjutkan ke karakter kedua dan seterusnya sampai semua karakter pada pola dan karakter pada teks dicocokkan sehingga terjadi kecocokan atau ketidakcocokan.

Saat $P_1 \dots P_m$ telah dicocokkan dengan $T_{i+1} \dots T_{i+m}$, Algoritma Quick Search menggunakan karakter pertama setelah karakter terakhir dari teks yang ada di dalam window sebagai patokan dari pergeseran pola selanjutnya yaitu karakter T_{i+m+1} , sehingga nilai tabel *bad character shift* untuk karakter P_m adalah 1. Inilah modifikasi tabel *bad character shift* untuk Algoritma Quick Search dari tabel *bad character shift* Boyer Moore. Bila karakter T_{i+m+1} tidak terdapat pada pola, maka pola akan digeser sebanyak $m+1$ ke kanan, sebaliknya bila karakter T_{i+m+1} terdapat pada pola maka karakter yang ada pada pola diposisikan sejajar/diluruskan dengan karakter T_{i+m+1} yang sama dan terkanan posisinya pada pola yang menjadi patokan pergeseran. Banyaknya pergeseran pola adalah $m+1-k$ dimana k adalah posisi karakter T_{i+m} pada pola.



(a) Karakter $T_{i+m+1} = O$ tidak terdapat pada pola, maka pola digeser sebanyak $m+1$

Posisi : 0 1 2 3 4 5 6 7 8

Teks :	A	L	G	O	<u>R</u>	I	T	M	A
Pola :	R	I	T						
Pola :					R	I	T		

Gambar 2

Pergeseran pola Algoritma Quick Search saat T_{i+m} tidak terdapat pada pola

Algoritma Quick Search sebagai berikut :

Input : Sebuah string T dengan panjang n, dan sebuah string P dengan panjang m

Output : Posisi ditemukannya P pada T ditampilkan ke layar

Procedure QS (T,P : string ; n,m :integer)

1. (Proses inialisasi array qs_bc mulai 0 sampai dengan maksimum alphabet dengan m+1)

Repeat for : 0 To ASIZE Do

- 1.1 (isi array qs_bc pada indeks i dengan m+1)

qs_bc (i) \leftarrow n + 1

2. (Perubahan isi array qs_bc sesuai dengan letak karakter dalam string P)

Repeat for i = 1 To n Do

- 2.1 (ganti isi array qs_bc pada indeks P[i] dengan m-i+1)

qs_bc (ord(P(i))) \leftarrow m-i+1

3. (Set indeks i ke awal teks)

i \leftarrow 0

4. (Proses pencocokan pola P dengan teks T. Ulangi selama indeks i belum melebihi n-m)

While i \leq n-m Do

- 4.1 (indeks j diset ke awal pola)

j \leftarrow 1

- 4.2 (proses pencocokan, ulangi selama karakter T(i+j) sama dengan karakter P(j) dan indeks j lebih kecil dari panjang pola)

While (j \leq m) **And** (T (i+j) = P(j)) Do

- 4.2.1 indeks j ditambah

- 4.3 (cek apakah pola ditemukan atau tidak)

If j > m

Then (pola ditemukan)

- 4.3.1 (cetak pergeseran pola yang dilakukan sehingga pola ditemukan dalam teks)

write (i)

(indeks i ditambah dengan nilai dari tabel qs_bc untuk karakter $T_{(i+m+1)}$)

$i \leftarrow i + qs_bc(\text{ord}(T_{(i+m+1)}))$

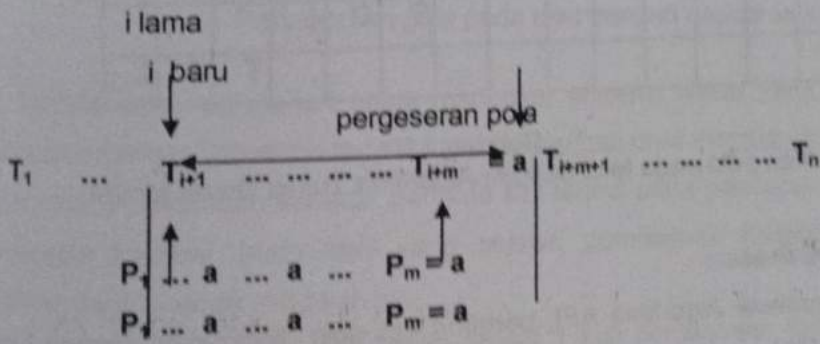
5. (Selesai)

Return

D. Modifikasi Algoritma Tuned Boyer Moore

Modifikasi Algoritma Tuned Boyer Moore hanya dilakukan pada pemilihan patokan pergeseran pola setelah terjadi pencocokan pola pada teks, dengan cara menggunakan pemilihan patokan pergeseran pola dari Algoritma Quick Search.

Pada Algoritma Tuned Boyer Moore, misalnya setelah pola $P_1 \dots P_m$ dicocokkan dengan teks $T_{i+1} \dots T_{i+m}$, kemudian pola digeser ke kanan dengan cara mensejajarkan antara karakter T_{i+m} sebagai patokan pergeseran pola dengan karakter dalam pola yang terkanan tetapi bukan karakter terakhir dimana karakter tersebut bernilai sama dengan T_{i+m} . Persyaratan 'bukan karakter terakhir dalam pola' untuk karakter yang sama dengan T_{i+m} , disebabkan jika karakter terakhir yang digunakan maka pola bisa saja tidak bergeser ke kanan.



Gambar 3

Karakter $T_{i+m} = 'a'$ sebagai patokan pergeseran pola, dan pola tidak bergeser ke kanan karena karakter dalam pola yang disejajarkan dengan $T_{i+m} = a$, adalah karakter yang paling kanan dan karakter terakhir dalam pola.

Dengan menggunakan Algoritma Tuned Boyer Moore, pertama kali pola dicocokkan dengan teks pada posisi ke-1 sampai ke-3. Karena karakter terakhir dari pola ('L') berbeda dengan karakter teks pada posisi ketiga ('O'), maka pola digeser sebanyak 3 kali. Kemudian terjadi pencocokan pola pada teks mulai posisi ke-3 sampai ke-5. Karena karakter terakhir dari pola ('L') sama dengan karakter teks pada posisi ke-5 ('L'), maka pencocokan dilanjutkan ke

karakter pertama dari pola ('T'), dan diperoleh ketidakcocokan dengan karakter teks pada posisi ke-3 ('O'). Lalu pola digeser ke kanan dengan cara menggunakan patokan pergeseran pola pada posisi ke-5 dari teks ('L') dan karakter 'L' yang terletak paling kanan dalam pola tetapi bukan karakter terakhir ternyata tidak terdapat, maka pola digeser melewati posisi ke-5 dan menempati posisi ke-6 sampai ke-8. Pergeseran pola hanya berlangsung sekali. Kembali pola dicocokkan dengan teks mulai posisi ke-6 sampai ke-8. Setelah pencocokan, pola digeser ke kanan dengan menggunakan patokan pergeseran pola pada posisi ke-8 yang juga huruf 'L'. Karena huruf 'L' ini tidak termuat pada bagian paling kanan tetapi bukan karakter terakhir dalam pola, maka pola digeser melewati karakter teks pada posisi ke-8, dan menempati posisi ke-9 sampai ke-11, demikian seterusnya.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
			∞											
Teks :	I	L	O	O	L	J	O	L	T	O	L	T	O	T
Pola :	T	O	L											
Pola :		T	O	L										
Pola :			T	O	L									
Pola :					T	O	L							
Pola :								T	O	L				
Pola :											T	O	L	

Gambar 5
Pencocokan pola pada teks dengan Algoritma Tuned Boyer Moore

E. Pengujian dan Pembahasan

Dengan menggunakan Algoritma AFI, pertama kali pola dicocokkan dengan teks pada posisi ke-1 sampai ke-3. Karena karakter terakhir dari pola ('L') berbeda dengan karakter teks pada posisi ketiga ('O'), maka pola digeser sebanyak 3 kali. Pada pergeseran pola yang pertama, digunakan patokan pergeseran pola pada posisi ke-4 dari teks yaitu huruf 'O'. Karena patokan ini termuat dalam pola maka pola digeser dengan cara mensejajarkan karakter 'O' pada pola di posisi ke-4. Pergeseran pola yang kedua menggunakan patokan pergeseran pola pada posisi ke-5 dari teks ('L'), berarti pola tidak bergeser. Pergeseran pola yang ketiga menggunakan patokan pergeseran pola pada posisi ke-5 dari teks ('L'), pada bagian ini juga pola tidak bergeser. Lalu dilakukan pencocokan pola pada teks posisi ke-3 sampai ke-5. Lalu pola digeser lagi sebanyak 1 kali, dengan menggunakan patokan pergeseran pola pada posisi ke-6 ('J'). Karena karakter 'J' tidak terdapat dalam pola maka pola digeser melewati posisi ke-6, dan menempati posisi ke-7 sampai ke-9. Pola dicocokkan lagi. Karena karakter terakhir dari pola ('L') tidak sama dengan

karakter teks pada posisi ke-9 ('T') maka terjadi 3 kali pergeseran pola ke kanan. Setelah pola digeser 3 kali, lalu terjadi pencocokan pada posisi ke-9 sampai ke-11, dan di posisi ini pola ditemukan. Pergeseran pola berlangsung 1 kali dengan menggunakan patokan pergeseran pola pada posisi ke-12 ('T'). Terakhir pola dicocokkan dengan teks pada posisi ke-12 sampai ke-14.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Teks :	I	L	O	O	L	J	O	L	T	O	L	T	O	T
Pola :	T	O	L											
Pola :			T	O	L									
Pola :			T	O	L									
Pola :			T	O	L									
Pola :							T	O	L					
Pola :									T	O	L			
Pola :									T	O	L			
Pola :												T	O	L

Gambar 6
Pencocokan pola pada teks dengan Algoritma AFI

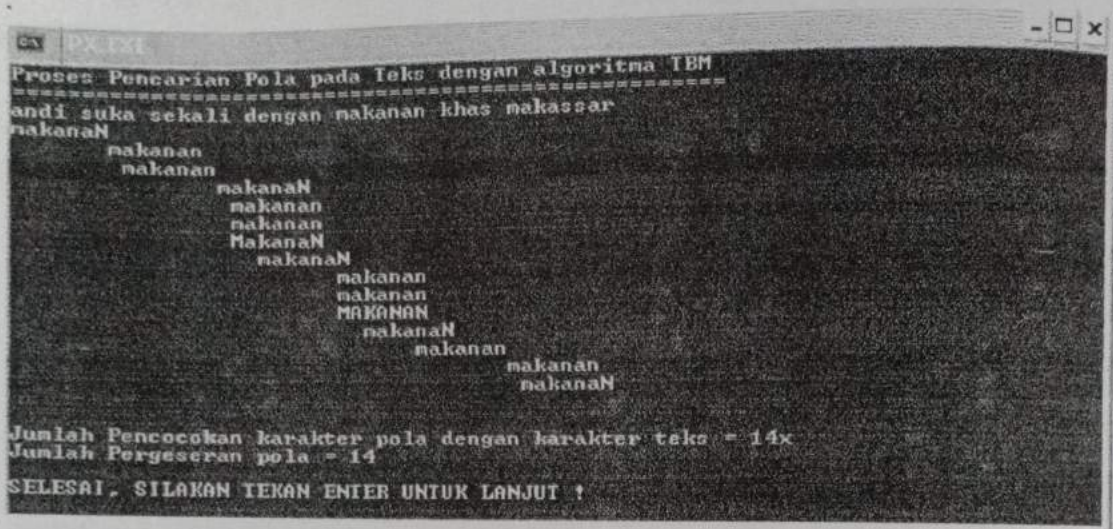
Metode yang digunakan untuk mengukur efisiensi waktu kerja antara Algoritma Tuned Boyer Moore dengan algoritma modifikasi yaitu dengan cara membandingkan jumlah pembacaan karakter teks. Pembacaan karakter pertama kali terjadi pada pencocokan karakter terakhir pada pola dengan karakter pada teks yang sejajar, pembacaan karakter dari teks selanjutnya didasarkan pada metode di bawah ini:

- a. Untuk pergeseran pola maka terjadi 1 kali pembacaan karakter teks yaitu patokan pergeseran pola.
- b. Pada pencocokan pola pada teks, dihitung berapa banyaknya karakter dari teks yang dibaca untuk dicocokkan dengan karakter dari pola.

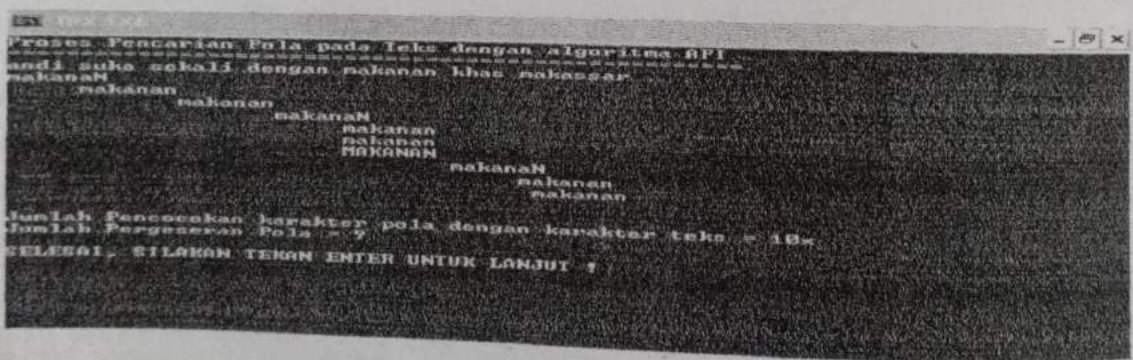
Pengujian terhadap kebenaran hasil implementasi kedua algoritma dengan memasukkan teks dan pola yang bemilai sembarang, kemudian memperhatikan apakah proses pencocokan dan pergeseran pola sudah benar untuk kedua algoritma. Jika terjadi kesalahan, maka dilakukan perbaikan hingga diperoleh proses pencocokan dan pergeseran pola yang benar. Pengujian terhadap perbandingan kedua algoritma dengan menghitung berapa jumlah pencocokan karakter teks dan karakter pola yang terjadi (x), dan berapa jumlah pergeseran pola

yang terjadi (y), kemudian nilai x dijumlahkan dengan nilai y, dan hasilnya adalah nilai frekuensi yaitu berapa kali terjadi pembacaan terhadap karakter teks selama proses pencocokan dan pergeseran pola terjadi.

Pengujian terhadap kebenaran hasil implementasi dari Algoritma Tuned Boyer-Moore dan Algoritma AFI sebagai berikut.



Gambar 7 Pencarian dengan Algoritma TBM dan AFI



Pada gambar di atas menunjukkan pencarian pola 'makanan' dengan menggunakan teks 'andi suka sekali dengan makanan khas makassar'. Pola dan teks yang diinput dibatasi pada huruf kecil, hal ini disebabkan huruf kapital digunakan untuk menunjukkan terjadinya perbandingan antara karakter pola dengan karakter teks. Sebagai contoh, pada gambar 7, pola 'makanan' dibandingkan dengan bagian awal dari teks yaitu 'andi su'. Yang pertama dibandingkan adalah karakter terakhir dari pola. Oleh karena itu karakter terakhir dari pola diubah ke huruf kapital.

Perbedaan utama Algoritma AFI dengan Algoritma Tuned Boyer-Moore, terletak pada penunjukan karakter pada teks yang menjadi patokan untuk pergeseran pola. Pada Algoritma Tuned Boyer Moore, untuk setiap kali setelah pencocokan pola pada teks, lalu pola digeser

dengan menggunakan patokan yang terletak pada karakter teks yang sejajar dengan karakter terakhir dari pola.

Contoh Tabel Pengujian

Percobaan	Panjang Pola	Panjang Teks	Jumlah Penocokan Karakter Pola dan Teks		Jumlah Pergeseran Pola		Frekuensi Pembacaan Karakter Teks	
			alg. TBM	alg. AFI	alg. TBM	alg. AFI	alg. TBM	alg. AFI
			1	10	1000	44	44	116
2	10	2000	88	88	226	226	314	314
3	10	3000	134	140	349	340	483	480
4	10	4000	178	178	450	447	628	625
5	10	5000	222	212	566	541	788	753

F. Kesimpulan

1. Modifikasi terhadap Algoritma Tuned Boye-Moore dilakukan pada pemilihan patokan pergeseran pola pada teks, dengan menggunakan cara pemilihan patokan pergeseran pola pada teks dari Algoritma Quick Search, yaitu menggunakan satu karakter berikut dari karakter teks yang sejajar dengan karakter terakhir dari pola.
2. Dari hasil implementasi dan pengujian, diperoleh gambaran bahwa Algoritma AFI lebih efisien dari Algoritma Tuned Boyer-Moore yang berpatokan pada frekuensi pembacaan karakter teks selama proses pencarian pola pada teks.
3. Untuk panjang teks yang semakin besar menunjukkan semakin efisiennya Algoritma AFI dibandingkan dengan Algoritma Tuned Boyer-Moore.

DAFTAR PUSTAKA

Cantone, D. Faro, S. 2003. <http://cs.felk.cvut.c2/psc/event/2003/p2.html>. Maret 2007

Charras, C., Lecroq, T. 1997. *Exact String Matching*. Universite de Rouen, France

<http://www.igm.univ-mlv.fr/~lecroq/string/tunedbm.htm>
 Februari 2007

ESMA.2000. <http://www4.kon.ne.jp/~felix/Algorithm/ESMA/Introduction.html>
 Februari 2007

Lecroq, T. 2001. *New Experimental Results on Exact String-Matching*.
<http://www.igm.univ-mlv.fr/~legroq>. Februari 2007.