

Perbandingan Algoritma Horspool dan Raita Pada Pencarian Data Mahasiswa

Siti Maesyaroh¹⁾, Daswa²⁾, Gentur Priguna Suwarto³⁾,

^{1,2,3)} Fakultas Ilmu Komputer, Universitas Kuningan

Jalan Pramuka No. 67, Purwawinangun Kecamatan Kuningan Kabupaten Kuningan, Jawa Barat 45512

Email : siti.maesyaroh@uniku.ac.id¹⁾, daswa@uniku.ac.id²⁾, gentur.priguna.suwarto@uniku.ac.id³⁾

Abstrak

Salah satu fitur yang terdapat di dalam sebuah aplikasi yaitu adanya fasilitas pencarian yang digunakan untuk memperoleh informasi. Jika data yang dikelola oleh suatu aplikasi terus bertambah dan memiliki kemiripan maka akan dapat mempengaruhi dalam proses pencarian untuk mendapatkan informasi yang tepat dan cepat. Pada sistem akademik di program studi Teknik Informatika FKOM UNIKU menyimpan data mahasiswa dengan jumlah 1003 mahasiswa dan data ini memungkinkan akan terus bertambah dengan adanya data mahasiswa baru sehingga saat pencarian membutuhkan waktu untuk mendapatkan informasi dikarenakan jumlah data yang banyak dan memiliki kesamaan pada identitas mahasiswa. Algoritma horspool dan algoritma raita merupakan algoritma pencocokan *string* yang dapat diterapkan dalam fitur pencarian untuk mendapatkan informasi tentang data mahasiswa yang dicari. Tujuan dari penelitian ini untuk membandingkan algoritma horspool dan algoritma raita berdasarkan waktu pencarian dan ketepatan informasi yang dicari. Oleh karena itu, penulis menerapkan algoritma horspool dan algoritma raita ke dalam sistem pencarian data mahasiswa. Kedua algoritma tersebut diterapkan pada halaman yang berbeda di sistem untuk mengetahui ketepatan informasi yang dicari berdasarkan informasi yang ditampilkan di sistem. Sedangkan untuk perbedaan waktu proses pencarian dari algoritma horspool dan algoritma raita dapat dilihat dari grafik perbandingan pada sistem. Hasil dari penelitian ini bahwa total rata-rata waktu pencarian untuk algoritma horspool adalah 9.268 ms dan algoritma raita adalah 4.307.

Kata Kunci : *pencocokan teks; mahasiswa; algoritma horspool; algoritma raita;*

Abstract

One of the features contained in an application is a search facility that is used to obtain information. If the data managed by an application continues to increase and is similar, it will influence the search process to get accurate and fast information. The academic system in Informatics Engineering study program FKOM UNIKU stores student data with a total of 1003 students and this that is likely to continue to increase with new student data so that when searching it takes time to get information because the amount of data is large and has similarities in student identity. The horspool algorithm and raita algorithm are string matching algorithms that can be applied in the search feature to obtain information about the student data being searched. The aim of this research is to compare the horspool algorithm and the raita algorithm based on search time and accuracy of the information sought. Therefore, the author applies the horspool algorithm and raita algorithm to the student data search system. These two algorithms are applied to different pages in the system to determine the accuracy of the information sought based on the information displayed in the system. Whereas, the difference in search processing time for the horspool algorithm and raita algorithm can be seen from the comparison graph on the system. The result of this research show that the average total search time for the horspool algorithm is 9.268 ms and the raita algorithm is 4.307 ms.

Keywords : *string matching; students; hosphool algorithm; raita algorithm;*

1. PENDAHULUAN

Di era digital ini pengelolaan data di suatu instansi telah memanfaatkan sebuah aplikasi untuk mempermudah pengguna dalam menyelesaikan pekerjaan. Salah satu fitur yang terdapat pada aplikasi pengelolaan data yaitu adanya sistem pencarian yang digunakan untuk mendapatkan informasi. Program studi Teknik Informatika Fakultas Ilmu Komputer Uniku telah menggunakan sistem akademik untuk mengelola data mahasiswa. Namun dengan jumlah data yang mencapai lebih dari 1000 mahasiswa dan memungkinkan akan terus bertambah dengan adanya data mahasiswa baru sehingga berpengaruh pada saat proses pencarian data yang membutuhkan waktu untuk mendapatkan informasi karena memiliki kesamaan pada biodata mahasiswa seperti, nama yang sama. Berdasarkan permasalahan tersebut diperlukan suatu metode pada sistem akademik yang dapat meningkatkan kinerja dalam mencari data mahasiswa agar memberikan informasi yang cepat dan tepat.

Penerapan algoritma pada fitur pencarian akan berpengaruh dalam proses pencarian data untuk mendapatkan informasi[1]. Algoritma merupakan langkah-langkah yang logis untuk menyelesaikan suatu permasalahan yang dapat diterapkan dalam sebuah sistem komputer. Oleh karena itu, sistem akademik di program studi Teknik Informatika FKOM Uniku pada fitur pencarian data diperlukan algoritma pencocokan teks (*string matching*) untuk meningkatkan kinerja pada proses pencarian data mahasiswa.

Dalam penelitian ini penulis menggunakan dua algoritma pencocokan teks yaitu algoritma horspool dan algoritma raita. Berdasarkan penelitian sebelumnya menyatakan bahwa algoritma horspool merupakan penyederhanaan dari algoritma boyer more yang dianggap sebagai algoritma pencarian string yang paling efisien (untuk aplikasi umum) saat ini, dikarenakan hanya menggunakan satu metode praproses yaitu *bad character shift*[2] sedangkan algoritma raita memiliki kecepatan yang baik dalam melakukan pencocokan *string*[1].

Pada tahun 2020 dengan judul penelitian yaitu perbandingan algoritma horspool dan algoritma *knuth morris pratt* pada aplikasi kamus farmasi berbasis android yang dilakukan oleh Ade Sitti Zainab, Sutardi, LM Tajidun, dan Rizal Adi Saputra dengan hasil penelitian bahwa algoritma *knuth morris pratt* lebih cepat melakukan pencarian dibandingkan algoritma horspool[3]. Pada tahun 2017 dengan penelitian yang berjudul perbandingan algoritma *binary search* dan raita dalam pencarian

data yang dilakukan oleh Dedi Rudi Bawanto dan Nidia Rosmawati[4]. Dari penelitian tersebut menghasilkan bahwa algoritma raita lebih cepat dari algoritma *binary search* dalam pencarian data dari keseluruhan data sampel yang diuji. Pada tahun 2021 dilakukan penelitian oleh Arya Kharismahendra dan Arliliwati tentang perbandingan dari algoritma *morris pratt* dan horspool algoritma pada term dari telekomunikasi aplikasi kamus berbasis android[5]. Hasil dari penelitian tersebut menunjukkan bahwa algoritma *morris pratt* lebih efisien dari algoritma horspool dan aplikasi dapat berjalan dengan baik. Pada tahun 2023 dilakukan penelitian oleh Herman Krismon Simbolon tentang perbandingan algoritma raita dan *apostolico crochemore* untuk pencarian data perpustakaan menggunakan metode eksponensial[1]. Dari hasil penelitian diketahui bahwa algoritma raita lebih cepat dari algoritma *apostolico crochemore*.

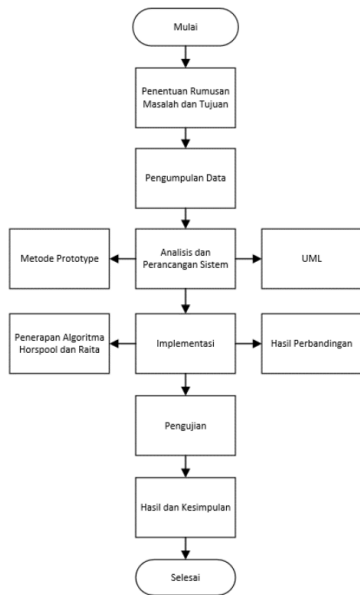
Setelah memaparkan hasil dari beberapa penelitian sebelumnya maka pada penelitian ini penulis membandingkan algoritma horspool dan algoritma raita berdasarkan kecepatan waktu dan ketepatan informasi yang dicari. Kedua algoritma tersebut diimplementasikan ke dalam sebuah sistem pada fitur pencarian data mahasiswa di program studi Teknik Informatika. Masing-masing algoritma diterapkan di halaman yang berbeda di sistem untuk mengetahui ketepatan informasi yang dicari berdasarkan informasi yang ditampilkan di sistem pada saat pencarian data. Sedangkan untuk kecepatan waktu pada saat proses pencarian data dapat dilihat di sistem dalam bentuk grafik. Satuan waktu yang digunakan untuk mengukur kecepatan pencarian data yaitu milidetik (ms). Hasil perbandingan algoritma yang diterapkan pada fitur pencarian data mahasiswa yaitu berupa perbedaan waktu pencarian data.

Penulis melakukan penelitian ini didasarkan pada pemahaman tentang kinerja algoritma pencocokan teks (*string matching*) untuk membandingkan algoritma horspool dan raita dalam pencarian data mahasiswa. Penelitian ini menggunakan data mahasiswa program studi Teknik Informatika sebanyak 1003 data dengan atribut yaitu nim, nama, kelas, dan status.

2. METODE PENELITIAN

Penelitian ini menggunakan deskriptif kualitatif karena berdasarkan data-data dari hasil observasi dan pengumpulan literatur untuk memperoleh informasi tentang algoritma pencocokan teks (*string matching*). Untuk alur kerja digambarkan dalam tahapan penelitian yang diawali

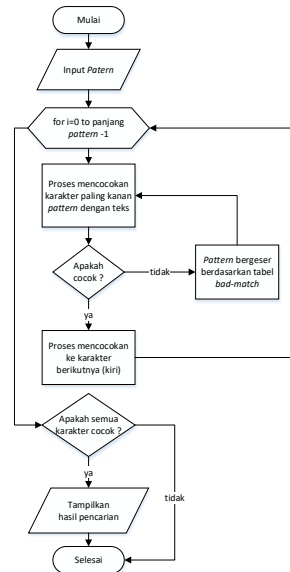
dari penentuan rumusan masalah sampai dengan hasil dan kesimpulan yang dapat dilihat pada gambar 1.



Gambar 1. Tahapan Penelitian

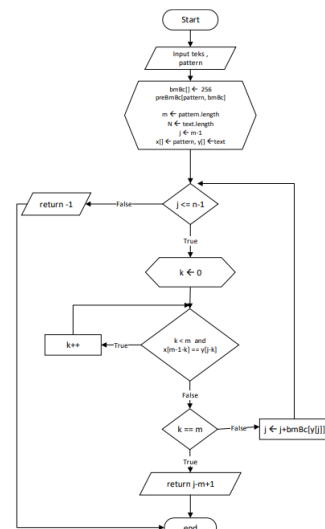
Pada tahap awal penelitian dimulai dengan menentukan rumusan masalah berdasarkan kondisi dan permasalahan yang terdapat di Program Studi Teknik Informatika. Penulis melakukan observasi dengan mengamati proses pencarian data mahasiswa pada sistem akademik. Hal ini dilakukan dengan mencari data mahasiswa menggunakan pencarian satu kata, dua kata dan tiga kata. Hasil dari pencarian tersebut dapat diketahui bahwa terdapat perbedaan dari informasi yang ditampilkan di sistem. Dimana, jika jumlah kata yang dimasukan semakin mendekati dengan data yang dicari maka sistem akan semakin sedikit menampilkan informasi yang dicari sehingga akan mudah untuk mendapatkan informasi sesuai yang dibutuhkan. Dari hasil observasi tersebut maka dapat dirumuskan suatu permasalahan sebagai bahan untuk melakukan penelitian ini.

Tahapan selanjutnya yaitu mengumpulkan data mahasiswa dan mempelajari beberapa literatur yang berkaitan dengan algoritma horspool dan algoritma raita. Penulis memperoleh data mahasiswa dari program studi Teknik Informatika yang digunakan pada sistem untuk diterapkan algoritma pencocokan teks. Pada bagian perancangan sistem menggunakan model diagram UML dengan metode prototype sebagai metode pengembangan sistem. Sedangkan untuk alur kerja sistem pada pencarian data mahasiswa dapat digambarkan dalam bentuk *flowchart* yang dapat dilihat pada gambar 2.



Gambar 2. Flowchart Sistem

Berdasarkan diagram alir sistem pada gambar 2 menggambarkan bahwa pengguna memasukkan teks berupa nama mahasiswa dengan satu kata kemudian sistem akan memproses pencarian nama untuk dicocokkan dengan *pattern* yang dimasukkan pada algoritma horspool maupun algoritma raita. Setelah itu, sistem akan menampilkan data mahasiswa yang sesuai atau cocok dengan *pattern*. Untuk proses pencarian data dimulai dari input teks dan *pattern* kemudian mencocokkan teks dengan *pattern* sehingga menghasilkan *output* berupa informasi tentang mahasiswa yang mengimplementasikan algoritma horspool dan raita yang dapat digambarkan dalam *flowchart* yang dapat dilihat pada gambar 3 untuk algoritma horspool dan gambar 4 untuk algoritma raita.

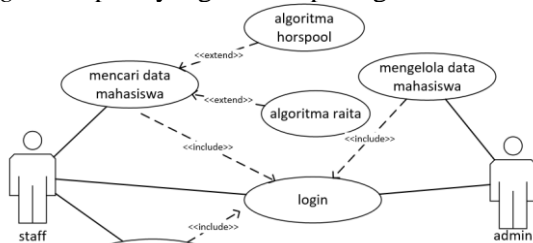


Gambar 3. Flowchart Algoritma Horspool



Gambar 4. Flowchart Algoritma Raita

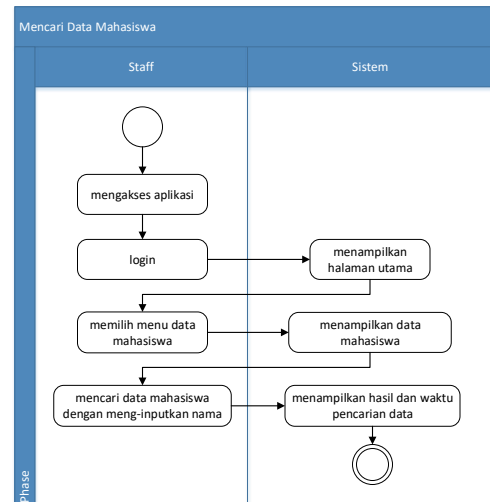
Dalam membangun sistem pencarian data mahasiswa yang menerapkan algoritma horspool dan algoritma raita pada tahap perancangan sistem digambarkan dalam bentuk *use case diagram*, *activity diagram*, dan *sequence diagram*. Sistem ini melibatkan dua aktor yang berinteraksi yaitu admin dan staff yang dapat digambarkan dalam *use case diagram* seperti yang terlihat pada gambar 5.



Gambar 5. Use Case Diagram

Pada gambar 5 menjelaskan bahwa aktor admin yang memiliki akses untuk mengelola data mahasiswa, seperti menambahkan data, mengubah data dan menghapus data sedangkan staff memiliki akses untuk mencari data mahasiswa yang telah diterapkan algoritma pencocokan teks dan melihat hasil perbandingan dari algoritma horspool dan algoritma raita.

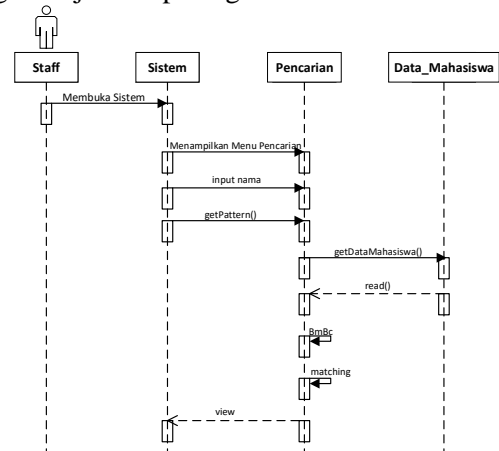
Activity diagram merupakan diagram yang menjelaskan bentuk alir kerja atau *workflow* atau suatu aktivitas pada sistem[6]. Untuk diagram aktivitas pencarian data mahasiswa terdapat pada gambar 6.



Gambar 6. Activity Diagram

Gambar 6 menjelaskan alur aktivitas saat pencarian data mahasiswa yang dilakukan oleh staff, dimana staff melakukan *login* ke sistem terlebih dahulu kemudian sistem akan menampilkan halaman utama. Selanjutnya staff memilih menu data mahasiswa dan melakukan pencarian data dengan meng-inputkan nama mahasiswa dengan satu kata. Sistem akan melakukan proses pencarian data menggunakan algoritma horspool dan raita yang sudah diterapkan pada sistem. Kemudian sistem menampilkan hasil pencarian yang dilengkapi dengan keterangan waktu pencarian pada masing-masing algoritma.

Sequence diagram yaitu suatu penyajian perilaku yang tersusun sebagai rangkaian langkah-langkah percontohan dari waktu ke waktu[7]. Adapun langkah-langkah pencarian data mahasiswa yang dilakukan oleh staff dengan menggunakan sistem dapat digambarkan dalam *sequence diagram* yang ditunjukkan pada gambar 7.



Gambar 7. Sequence Diagram

Gambar 7 menunjukkan kegiatan yang dilakukan oleh staff saat berinteraksi dengan sistem untuk mencari data mahasiswa. Diawali dengan staff masuk ke sistem kemudian mengakses fitur pencarian data. Staff mengetikkan nama mahasiswa pada kolom pencarian. Setelah itu sistem akan mencocokkan nama mahasiswa yang dicari dengan nama yang ada di *database* dan menghasilkan *output* berupa informasi yang berkaitan dengan mahasiswa tersebut.

Algoritma horspool dan algoritma raita diterapkan pada fitur pencarian data berupa sistem berbasis web untuk memperoleh informasi tentang mahasiswa. Algoritma horspool melakukan pencocokan *string* dimulai dari karakter paling kanan dari *pattern* yang akan dicocokkan dengan teks. Untuk melewati proses ketika terjadi ketidakcocokan antara karakter *pattern* dengan karakter teks, maka menggunakan *bad-match* secara berulang sampai seluruh karakter *pattern* mempunyai kecocokkan terhadap teks. Pencocokkan *string* menggunakan algoritma horspool terdapat 2 (dua) tahap, yaitu tahap *pra processing* dan tahap *searching*[3].

a. Tahap *pra processing*

Tahap *pra processing* meliputi observasi *pattern* terhadap teks untuk membuat sebuah tabel *bad-match* yang berisi nilai *shift* jika terjadi ketidakcocokkan antara *pattern* dan teks. Berikut ini langkah-langkah yang dilakukan algoritma *horspool* pada tahap praproses, yaitu sebagai berikut :

1. Algoritma *horspool* melakukan pencocokan karakter terkanan pada *pattern*.
2. Setiap karakter pada *pattern* ditambah ke dalam tabel *bad-match* dan dihitung nilai *shift*-nya.
3. Karakter yang berada pada ujung *pattern* tidak dihitung dan tidak dijadikan karakter terkanan dari karakter yang sama dengannya.
4. Jika terdapat dua karakter yang sama dan salah satunya bukan karakter terkanan, maka karakter dengan indeks terbesar yang dihitung nilai *shift*-nya.
5. Algoritma *horspool* menyimpan panjang dari *pattern* sebagai panjang nilai *shift* secara *default* jika karakter pada teks tidak ditemukan dalam *patern*.
6. Nilai (BM) *shift* yang akan digunakan dapat dicari dengan perhitungan panjang dari *pattern* dikurang indeks terakhir karakter dikurang 1, untuk masing-masing karakter (BC).

$$BM = m - i - 1 \tag{1}$$

Sebagai contoh terdapat *pattern* : RIZKY

R I Z K Y
0 1 2 3 4

Dengan menggunakan rumus (1) maka dapat dibuat nilai *bad-match* pada tahap praproses seperti pada tabel 1.

Tabel 1. *Bad-Match* Pada Praproses

Karakter	Indeks	Nilai
R	0	4
I	1	3
Z	2	2
K	3	1
Y	4	0
*	-	5

Nilai = 5 - 0 - 1 = 4

Nilai = 5 - 1 - 1 = 3

Nilai = 5 - 2 - 1 = 2

Nilai = 5 - 3 - 1 = 1

Nilai = 5 - 4 - 1 = 0

* : karakter yang tidak dikenali

Algoritma horspool yang diterapkan dalam penelitian ini sebagai contoh untuk mencocokkan *patern* yaitu RIZKY dengan teks yaitu MUHAMMAD RIZKY FAUZI dengan inisialisasi awal yang ditunjukkan pada tabel 2.

Tabel 2. Inisialisasi awal

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
T	M	U	H	A	M	M	A	D		R	I	Z	K	Y		F	A	U	Z	I
P	R	I	Z	K	Y															
i	0	1	2	3	4															

Sedangkan untuk nilai *bad-match* berdasarkan rumus (1) dapat dibuat seperti pada tabel 3.

Tabel 3. Pembuatan *Bad-Match*

P	R	I	Z	K	Y	*
I	0	1	2	3	4	-
V	4	3	2	1	0	5

Tabel 2 menunjukkan pemberian nilai awal untuk setiap teks dan *pattern*. Teks dan *pattern* diberi nilai m dan i dengan nilai m sebagai panjang *pattern* dan nilai i sebagai *indeks*. Sedangkan tabel 3 menunjukkan pergeseran nilai *bad-match* dengan menghitung nilai v seperti yang telah dilakukan pada tabel 2.

b. Tahap *searching* (pencarian)

Pada proses pencarian dilakukan dengan membandingkan karakter paling kanan pada *pattern* dengan teks. Jika terjadi

ketidakcocokan, akan dilakukan pergeseran ke kanan dengan melewati karakter yang tidak cocok dimana nilai pergeserannya berdasarkan pada tabel *bad-match*. Karakter paling kanan teks berfungsi sebagai landasan untuk menentukan jarak geser yang akan dilakukan. Untuk tahap pencarian diawali seperti yang ditunjukkan pada tabel 4.

Tabel 4. Iterasi Algoritma Horspool Pertama

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
T	M	U	H	A	M	M	A	D		R	I	Z	K	Y		F	A	U	Z	I
P	R	I	Z	K	Y															
i	0	1	2	3	4															

Iterasi pertama pada tabel 4 menunjukkan bahwa terdapat ketidakcocokkan karakter “M” dengan “Y”. Karakter “M” tidak terdapat pada tabel *bad-match* sehingga diganti dengan tanda (*). Tanda (*) pada tabel *bad-match* bernilai 5 maka dilakukan pergeseran sebanyak 5 kali yang hasilnya dapat dilihat pada tabel 5.

Tabel 5. Iterasi Algoritma Horspool Kedua

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
T	M	U	H	A	M	M	A	D		R	I	Z	K	Y		F	A	U	Z	I
P						R	I	Z	K	Y										
i						0	1	2	3	4										

Iterasi kedua yang terdapat pada tabel 5 memiliki ketidakcocokkan antara karakter “R” dengan “Y”. Karakter “R” pada tabel *bad-match* bernilai 4 sehingga dilakukan pergeseran sebanyak 4 kali seperti yang terlihat pada tabel 6.

Tabel 6. Iterasi Algoritma Horspool Ketiga

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
T	M	U	H	A	M	M	A	D		R	I	Z	K	Y		F	A	U	Z	I
P										R	I	Z	K	Y						
i										0	1	2	3	4						

Pada iterasi ketiga, karakter yang terdapat pada teks dengan *pattern* sudah sesuai dan semua *pattern* memiliki kecocokkan dengan teks. Dengan demikian, proses pencarian *pattern* dengan teks menggunakan algoritma horspool berakhir pada iterasi ketiga.

Berbeda dengan algoritma raita yang membandingkan karakter yang terakhir dari pola teks paling kiri. Jika memiliki kecocokkan, maka karakter pertama dari pola teks paling kiri juga dibandingkan. Kemudian membandingkan karakter tengah pola dengan karakter teks tengah. Jika semuanya memiliki kecocokkan, maka algoritma membandingkan karakter lain mulai dari karakter kedua ke karakter kedua terakhir dan membandingkan dengan karakter tengah lagi[8].

Untuk proses pencarian menggunakan algoritma raita memiliki beberapa tahapan, diantaranya yaitu :

- Membuat tabel pergeseran pola yang dicari sebagai kata yang akan dicari pada teks.
- Jika dalam proses membandingkan antara pasangan karakter pada akhir pola dengan karakter teks terjadi ketidakcocokkan, maka pergeseran dilakukan sesuai nilai karakter pada tabel BmBc.
- Jika dalam proses membandingkan akhir pola terjadi ketidakcocokkan lagi maka karakter akan digeser lagi sesuai dengan nilai yang terdapat di tabel BmBc.
- Jika karakter akhir pola dengan karakter pada teks yang sedang dibandingkan cocok, maka posisi karakter pada pola dan teks akan memiliki nilai (0), dan dilanjutkan pencocokkan pada karakter awal pola. Jika cocok maka dilanjutkan pencocokkan dengan karakter tengah pola.
- Jika akhir, awal, dan tengah pola sudah cocok maka pencocokkan dilanjutkan dengan bagian kanan dari awal karakter pada pola, jika cocok maka selanjutnya mencocokkan pada bagian kanan tengah pola.

Pada penelitian ini penulis menerapkan algoritma raita pada teks dan *pattern* yaitu :

Teks = MUHAMMAD RIZKY FAUZI

Pattern = RIZKY

Untuk melakukan perhitungan maka dibuat tabel BmBc dengan rumus :

$$k = m - 2 \tag{2}$$

Kemudian menghitung batas pencarian karakter pada pola.

$$BM[i] = m - i - 1 \tag{3}$$

Tabel BmBc digunakan sebagai pencari nilai karakter seperti yang terlihat pada tabel 7.

R	I	Z	K	Y
↓	↓	↓	↓	↓
0	1	2	3	4

Tabel 7. Nilai BmBc

Bc	Bm
R	4
I	3
Z	2
K	1
Y	5
*	5

Tabel 7 dapat diperoleh dari perhitungan nilai BmBc yaitu :

$$k = m - 2 = 5 - 2 = 3 \text{ maka } i = 0 - 3$$

Untuk mencari nilai pergeseran pada tabel BmBc menggunakan perhitungan seperti berikut :

$$Bm [Bc[R]] = 5 - 0 - 1 = 4$$

$$Bm [Bc[I]] = 5 - 1 - 1 = 3$$

$$Bm [Bc[Z]] = 5 - 2 - 1 = 2$$

$$Bm [Bc[K]] = 5 - 3 - 1 = 1$$

Nilai untuk karakter Y adalah sesuai dengan panjang pola yaitu 5.

(*) = karakter yang tidak dikenal

Jumlah karakter pada pola sebanyak 5 sehingga untuk karakter yang tidak ada pada pola diinisialisasikan dengan tanda (*) yang nilainya sesuai dengan jumlah pola (m).

Proses pencarian data menggunakan algoritma raita diawali dengan mencocokkan karakter akhir pola dengan teks seperti yang terlihat pada tabel 8, jika terjadi ketidakcocokkan maka karakter pola akan bergeser ke kanan sebanyak nilai teks.

Tabel 8. Pencarian Teks Proses Pertama

T	M	U	H	A	M	M	A	D		R	I	Z	K	Y		F	A	U	Z	I
P	R	I	Z	K	Y															

Pencarian pertama pada tabel 8 terdapat ketidakcocokan antara karakter “M” dan “Y”. Karakter “M” tidak terdapat pada tabel BmBc sehingga digantikan dengan tanda (*). Tanda (*) memiliki nilai 5 maka pergeseran dilakukan sebanyak 5 kali. Oleh karena itu, pencarian selanjutnya yaitu *pattern* bergeser sebanyak 5 kali seperti yang ditunjukkan pada tabel 9 yang merupakan proses kedua.

Tabel 9. Pencarian Teks Proses Kedua

T	M	U	H	A	M	M	A	D		R	I	Z	K	Y		F	A	U	Z	I	
P							R	I	Z	K	Y										

Pencarian kedua pada tabel 9 terdapat ketidakcocokkan antara karakter “R” dan “Y”. Karakter “R” pada tabel BmBc memiliki nilai 4 sehingga pencarian tahap ketiga dilakukan pergeseran sebanyak 4 kali seperti yang tertulis pada tabel 10.

Tabel 10. Pencarian Teks Proses Ketiga

T	M	U	H	A	M	M	A	D		R	I	Z	K	Y		F	A	U	Z	I	
P										R	I	Z	K	Y							

Karakter akhir pola dengan karakter akhir teks pada tabel 10 terjadi kecocokkan, maka selanjutnya dilakukan pencocokan pada karakter awal pola dan karakter tengah pola dengan teks. Dari tabel 10

menunjukkan adanya kecocokkan untuk semua bagian pola dengan teks. Dengan demikian, proses pencarian teks menggunakan algoritma raita sudah selesai pada pencarian teks tahap ketiga.

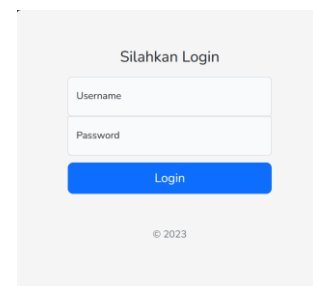
Sedangkan untuk mengetahui kecepatan waktu sebuah algoritma dapat diukur dengan menghitung jumlah instruksi yang dieksekusi. Setiap instruksi dapat dihitung dengan besaran waktu. Setelah mempelajari beberapa referensi diketahui bahwa untuk menghitung kompleksitas waktu pada algoritma horspool dan algoritma raita menggunakan besaran waktu yaitu satuan milidetik. Pada penelitian ini menggunakan perhitungan kompleksitas waktu algoritma horspool dan raita berdasarkan kebutuhan waktu saat pencarian data secara rata-rata.

Rumus untuk menghitung waktu rata-rata terdapat pada persamaan (4) dengan $T_{avg}(n)$ merupakan kompleksitas waktu untuk kasus rata-rata (*average case*). Dimana, n adalah jumlah instruksi/operasi yang dieksekusi oleh algoritma.

$$T_{avg}(n) = \frac{(1+2+3+\dots+n)}{n} = \frac{\frac{1}{2}n(1+n)}{n} = \frac{(n+1)}{2} \quad (4)$$

3. HASIL DAN PEMBAHASAN

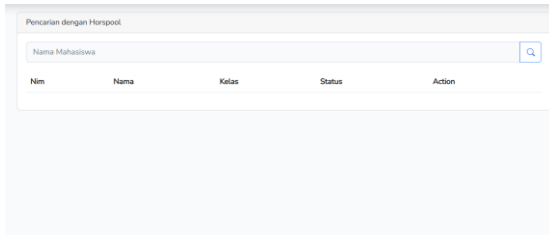
Hasil implementasi pada penelitian ini berupa sistem pengelolaan data mahasiswa yang dilengkapi dengan fitur pencarian data yang menerapkan algoritma horspool dan algoritma raita. Pada saat aplikasi dijalankan maka tampilan awal yaitu halaman form login yang ditunjukkan pada gambar 8.



Gambar 8. Halaman Login

Halaman login dapat digunakan oleh admin dan staff. Admin bertugas untuk mengelola data mahasiswa, seperti menambahkan, mengubah, dan menghapus data sedangkan staff dapat melakukan pencarian data mahasiswa menggunakan algoritma horspool dan algoritma raita.

Staff melakukan pencarian data pada menu pencarian untuk melakukan pencarian data mahasiswa yang menerapkan algoritma horspool dan algoritma raita seperti pada gambar 9.



Gambar 9. Halaman Pencarian

Setelah meng-inputkan nama yang dicari selanjutnya sistem akan menampilkan data mahasiswa seperti yang terlihat pada gambar 10.

Nim	Nama	Kelas	Status	Action
20160010102	Adi Rizky Karana	TIMC 2016-02	Absen	[X]
20160010138	Rizky Fauzan Humala	TIMC 2016-02	Absen	[X]
20160010096	Rizky Widayanda	TIMC 2016-03	Absen	[X]
20170010037	Angga Rizky Nur Perunggan	TIMC 2017-04	Lulus	[X]
20170010024	Igal Rizky	TIMC 2017-03	Absen	[X]
20170010045	Rizky Dendarmawan	TIMC 2017-02	Absen	[X]
20180010083	Alsy Rizky Junar	TIMC 2018-01	Absen	[X]
20180010073	Diova Rizky Analia	TIMC 2018-02	Lulus	[X]
20220010051	Adhys Rizky Rahayu	TIMC 2022-03	Absen	[X]
20220010165	Muhammad Rizky Faat	TIMC 2022-04	Absen	[X]
20220010006	Rizky Ramana Supatman	TIMC 2022-04	Absen	[X]

Gambar 10. Halaman Hasil Pencarian Data

Gambar 10 merupakan halaman yang menampilkan hasil pencarian dari satu kata yang di-inputkan oleh staff dengan menampilkan semua data mahasiswa yang mengandung kata tersebut. Seperti contoh yang terdapat pada gambar 9 dimana staff memasukkan kata “Rizky” maka sistem akan mencari dan menampilkan informasi mahasiswa yang terdapat kata tersebut.

Hasil pencarian menggunakan algoritma horspool dan algoritma raita dengan menggunakan satu kata pada masing-masing nama mahasiswa diperoleh hasil rata-rata waktu untuk algoritma horspool adalah 9.268 ms sedangkan algoritma raita adalah 4.307 ms. Hasil tersebut berdasarkan pada pengujian sebanyak 16 data dengan menggunakan satu kata seperti pada gambar 11.



Gambar 11. Hasil Pengujian

Untuk hasil perbandingan antara algoritma horspool dan algoritma raita berdasarkan rata-rata waktu pencarian data yang digambarkan dalam bentuk grafik pada gambar 12.



Gambar 12. Hasil Perbandingan

3.1. Pengujian (Testing)

Pengujian difokuskan kepada implementasi algoritma untuk memeriksa ketepatan informasi dan kecepatan waktu pencarian data mahasiswa yang dilakukan oleh staff. Pada penelitian ini untuk pengujian hanya dilakukan dengan meng-inputkan satu kata dari nama mahasiswa yang akan dicari ke dalam form pencarian terhadap sistem yang dibangun. Tabel 11 menunjukkan hasil pengujian menggunakan algoritma horspool dengan satu kata.

Tabel 11. Pengujian Algoritma Horspool

No.	Kata yang dicari	Pengujian Algoritma Horspool ke-															Rata-rata	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		16
1	Abdul	12.3	9.5	9.4	11.2	10.7	11.5	12.9	12.1	12.9	11	11.4	11	12.5	12	13.3	11.9	11.60
2	Agung	11.8	12.8	11.3	12.7	13.6	9.7	11.7	12.4	11.5	9.3	10.9	9.2	11.2	10.9	10.1	12.5	11.35
3	Aspi	11.4	12.9	10.6	11	10.8	8.3	10.5	10.2	12.3	11.6	8.3	9.5	8.4	10.7	10.7	9.4	10.41
4	Cici	11.9	10.7	9.7	10.8	13.1	11.3	9.8	9.7	10.9	9.5	10.5	9.7	10.3	11.6	7.2	10.3	10.44
5	Dhaka	8.7	7.9	6	9.8	7.9	5.4	7.5	7.7	7.9	5.4	7.6	6.7	6.7	5.6	7.3	7.8	7.24
6	Dwi	12.3	9.9	8.9	11.7	11.4	8.5	8.3	8.8	8.5	11.6	10.4	10.1	7.8	7.8	9.2	11.1	9.74
7	Egi	14.8	12.6	12.3	12.6	13.8	10.9	11.2	12.8	11.5	13.4	12.3	12.3	12.9	11.2	11.5	12.2	12.39
8	Fajar	11.2	7.9	9.5	11.8	10.8	8.6	7.8	9.7	7.6	7.8	7.6	10.1	8.9	9.2	9.8	7.6	9.12
9	Gilang	9.7	9.5	11.4	9.9	10.5	9.3	8.3	11	9.3	10.1	10.2	8.9	7.5	7.8	10.6	7.8	9.49
10	Imam	8.1	7.4	7.2	10	8.9	9.1	7.7	8.8	8.9	7.8	6.2	6.6	8.9	8.9	8.3	6.9	8.11
11	Maulana	12.4	11.3	12.5	13.9	15.1	11.9	10.1	11.1	11.5	10.2	10.5	12.3	12.1	10.2	13.1	10.9	11.82
12	Putri	5.9	4.9	4.5	6.3	4.9	4.7	4.3	5.6	3.5	4.4	6.1	4.7	4.5	5.6	4.3	3.5	4.86
13	Reza	11.3	9.5	10.9	11.8	10.8	8.5	7.2	9.8	9.1	10.5	9.5	8.9	7.8	7.9	8.6	8.7	9.43
14	Rizki	9.3	9.9	8.4	10.7	8.8	8.6	6.9	8.3	7.3	6.7	7.7	6.8	9	8.1	7.3	6.9	8.17
15	Rizal	6.7	4.8	5.3	6.9	4.6	4.9	4.4	4.5	3.4	5.5	5.3	4.9	3.9	5.4	4.2	6.9	5.10
16	Rizky	11.5	9.1	9.5	11.4	9.8	9.3	7.8	8.7	9.3	8.3	7.9	7.5	8.3	10	7.9	8.5	9.04
		Rata-rata															9.268	

Sedangkan tabel 12 menunjukkan hasil pengujian menggunakan algoritma raita dengan satu kata.

Tabel 12. Pangujian Algoritma Raita

No.	Kata yang dicari	Pengujian Algoritma Raita ke-															Rata-rata	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		16
1	Abdul	5.2	6.3	5.7	4.7	5.2	5.3	6.8	5.8	5.1	4.6	5.2	4.9	5.1	5.2	5.1	5.2	5.34
2	Agung	6.8	5.7	5.2	5.3	6.2	6.5	5.5	5.4	5.7	6.1	5.5	5.4	5.4	3.8	3.8	3.1	5.34
3	Aspi	5.6	5.7	5.1	4.4	5.3	4.1	4.3	3.8	5.2	4.2	3.8	4.3	5.2	3.5	3.3	3	4.43
4	Cici	5.3	5.7	4.3	4.6	5.7	4.1	4.6	3.3	3.8	5.7	4.2	4.3	5.1	4.8	4.9	4.2	4.66
5	Dhaka	6.7	5.6	5.7	6.3	6.4	6.5	5.2	5.2	4.7	4.1	3.9	3.3	5.3	4.8	4.8	6.3	5.30
6	Dwi	6.3	6.1	5.5	5.2	4.8	5.3	3.5	4.1	3.2	3.2	5.2	5.7	3.3	2.7	4.7	3.6	4.53
7	Egi	6.5	5.3	4.4	2.4	5.3	2.7	3.1	4.9	4.2	4.6	3.1	5.3	5.7	4.9	5.1	6.2	4.61
8	Fajar	5.7	4.3	3.9	1.6	3.3	2.2	2.6	2.5	2.4	2.5	2	3.8	3.7	2.6	5.7	4.8	3.35
9	Gilang	6.4	6.3	5.1	6.7	5.7	6.1	6.5	5.9	4.6	4.8	6.5	4.2	5.2	4.8	4.8	5.2	5.55
10	Imam	5.3	6.7	6.1	3.8	2.3	2.3	3.2	2.4	2.6	2.3	2.7	3.4	2.1	4.6	2.6	2.3	3.42
11	Maulana	5.7	5.2	6.5	5.3	5.8	4.7	5.1	4.7	6.1	5.4	6.4	6.1	4.9	4.9	5.1	4.1	5.98
12	Putri	6.3	6.7	5.6	6.7	6.1	4.9	3.4	4.2	5.2	4.5	4.6	5.3	4.2	3.8	3.8	3.2	4.31
13	Reza	3.5	2.7	1.4	3.3	2.3	2.3	2.3	2.3	3.3	2.3	4.7	4.7	2.4	2.6	2.6	3.3	2.88
14	Rizki	5.8	5.3	4.4	2.2	2.9	5.4	4.7	2.5	2.4	5.2	4.2	2.6	2.2	2.6	2.8	2.6	3.61
15	Rizal	5.5	5.5	3.3	3.6	3.6	4.7	3.1	3.1	2.5	2.3	2.1	2.3	2.3	2.2	2.1	2.7	3.18
16	Rizky	3.9	2.3	1.7	2.2	2.6	2.1	2.3	2.3	2.4	2.5	2.8	2.3	2.5	2	2.5	2.8	2.45
		Rata-rata															4.307	

Hasil pengujian algoritma dari tabel 11 dan tabel 12 menyatakan bahwa pencarian data mahasiswa menggunakan satu kata diperoleh rata-rata waktu untuk algoritma horspool yaitu 9.268 ms sedangkan algoritma raita yaitu 4.307 ms. Dengan demikian, hasil tersebut menunjukkan bahwa algoritma raita melakukan pencarian teks lebih cepat

dibandingkan algoritma horspool dengan selisih waktu yaitu 4.961 ms.

4. KESIMPULAN

Berdasarkan pembahasan yang telah diuraikan mengenai perbandingan algoritma horspool dan algoritma raita yang diterapkan pada sistem berbasis web untuk fitur pencarian data mahasiswa maka kesimpulan dari penelitian ini yaitu algoritma horspool dan algoritma raita memiliki persamaan yaitu melakukan pencocokan karakter yang diawali pada bagian paling kanan dari teks.

Total rata-rata kecepatan waktu dalam pencarian teks menggunakan satu kata untuk algoritma horspool yaitu 9.268 ms, sedangkan untuk algoritma raita yaitu 4.307 ms. Dari hasil perbandingan kecepatan waktu tersebut maka algoritma raita melakukan pencarian teks lebih cepat dibandingkan dengan algoritma horspool.

5. SARAN

Berdasarkan hasil penelitian ini maka dapat disampaikan beberapa saran yang dapat digunakan untuk pengembangan penelitian selanjutnya, yaitu :

- a. Untuk pengujian algoritma dapat diterapkan dengan menggunakan dua kata atau lebih.
- b. Algoritma yang digunakan pada penelitian ini dapat dikembangkan dengan algoritma yang lain.

UCAPAN TERIMA KASIH

Kami mengucapkan terima kasih kepada pihak yang telah banyak membantu dalam penelitian ini yaitu kepada Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Kuningan dan LPPM Universitas Kuningan atas dukungan material maupun non material.

DAFTAR PUSTAKA

- [1] H. K. Simbolon, "Bulletin of Information System Research (BIOS) Perbandingan Algoritma Raita Dan Apostolico Crochemore Untuk Pencarian Data Perpustakaan Menggunakan Metode Eksponensial," 2023, [Online]. Available: <https://journal.grahamitra.id/index.php/bios>
- [2] D. Wahyuni Kartini, P. Algoritma Boyer Moore Horspool Pada Pencarian Katalog Buku Dian wahyuni Kartini, R. Yanur Tanjung, and I. Lasmana, *Seminar Nasional Sains & Teknologi Informasi (SENSASI)*. 2019. [Online]. Available: <http://prosiding.seminar-id.com/index.php/sensasi/issue/archivePage|341>
- [3] P. Algoritma Horspool Dan Algoritma Knuth Morris Pratt Pada Aplikasi Kamus Farmasi Berbasis Android, A. Sitti Nur Zainab, L.

- Tajidun, R. Adi Saputra, and J. Teknik Informatika, "Terakreditasi 'Peringkat 4 (Sinta 4)' oleh Kemenristekdikti This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License," vol. 6, no. 1, pp. 1–5, doi: 10.5281/zenodo.3856171.
- [4] D. Rudi Bawanto and N. Rosmawanti, "Perbandingan Algoritma Binary Search Dan Raita Dalam Pencarian Data".
- [5] A. Kharismahendra and C. Author, "Sintaksis : Jurnal Ilmiah Pendidikan PERBANDINGAN DARI ALGORITMA MORRIS-PRATT DAN HORSPOOL ALGORITMA PADA TERM DARI TELEKOMUNIKASI APLIKASI KAMUS BERBASIS ANDROID".
- [6] A. Azhar, N. Marbun, S. Aripin, and E. Buulolo, "IMPLEMENTASI ALGORITMA HORSPOOL PADA APLIKASI ISTILAH FASHION," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 3, no. 1, Nov. 2019, doi: 10.30865/komik.v3i1.1641.
- [7] S. Maesyaroh, N. A. Asikin, and D. Daswa, "INNOVATION OF WEB BASED ENGLISH LEARNING MEDIA DEVELOPMENT," *English Review: Journal of English Education*, vol. 11, no. 2, pp. 451–460, Jun. 2023, doi: 10.25134/erjee.v11i2.7468.
- [8] N. Marbun, M. Zarlis, D. Hartama, and B. J. Sitompul, *Seminar Nasional Sains & Teknologi Informasi (SENSASI) Implementasi Algoritma Raita Pada Pencarian Katalog Alkes*. [Online]. Available: <http://prosiding.seminar-id.com/index.php/sensasi/issue/archivePage|520>