

Perbandingan Algoritma *Binary Search* Dan *Raita* Dalam Pencarian Data

Dedi Rudi Bawanto¹, Nidia Rosmawanti²

Program Studi Sistem Informasi, STMIK Banjarbaru

Jl. A. Yani Km. 33,3 Banjarbaru, Telp (0511) 4782881

¹dedirudibawanto@gmail.com, ²nidiabjb@yahoo.com

Abstrak

Aplikasi komputer merupakan salah satu produk teknologi yang membantu manusia dalam berbagai pekerjaan, dan pencarian data merupakan kegiatan yang sering dilakukan oleh manusia sebagai pengguna komputer. Dengan jumlah data yang banyak dan terus bertambah serta algoritma pencarian yang digunakan sangat berpengaruh terhadap waktu pencarian data, sehingga fasilitas pencarian dengan algoritma lebih cepat dan tepat menjadi wajib pada sebuah sistem atau aplikasi komputer. Dari penelitian ini menyimpulkan bahwa ditinjau dari total waktu pencarian setiap algoritma dengan teknik pengujian yaitu match case, part case, 32 record data, 64 record data, 96 record data, tipe data string, tipe data *numeric* dan tipe data *date* dengan 5 kali simulasi dari setiap kata kunci pada teknik pengujian, algoritma *raita* lebih cepat 3% dari algoritma *binary search* dalam pencarian data dari keseluruhan data sampel yang diuji dengan total waktu 2.5116 ms dibandingkan penelitian terdahulu yang mana algoritma *boyer moore* dengan total waktu 2.8150 ms lebih lambat 1% dari algoritma *binary search* dengan total waktu 2.7830 ms.

Kata Kunci: *Perbandingan, Algoritma Raita, Algoritma Binary Search.*

Abstract

Computer applications is one of the technology products that help people in a variety of jobs, and data retrieval is an activity that is often done by humans as computer users. With the amount of data is large and growing, and the search algorithm used is very influential on search time data, so the search algorithm more quickly and precisely became mandatory in a computer system or application. From this study it can be concluded that in terms of total search time of each algorithm by testing techniques that match the case, part case, 32 data records, 64 data records, 96 records of data, string data types, data type numeric and data types date with 5 times the simulation of each keyword in the testing techniques, raita algorithm 3% faster than binary search in the search data of all the data samples tested for a total of 2.5116 ms compared to previous studies where the Boyer-Moore string search algorithm with a total time of 2.8150 ms 1% slower of binary search algorithm with a total time of 2.7830 ms.

Keywords: *Comparison, Raita algorithm, binary search algorithm.*

1. Pendahuluan

Aplikasi komputer merupakan salah satu produk teknologi yang membantu manusia dalam berbagai pekerjaan, dan pencarian data merupakan kegiatan yang sering dilakukan oleh manusia sebagai pengguna komputer. Dengan jumlah data yang banyak dan terus bertambah serta algoritma pencarian yang digunakan sangat berpengaruh terhadap waktu pencarian data, sehingga fasilitas pencarian dengan algoritma lebih cepat dan tepat menjadi wajib pada sebuah sistem atau aplikasi komputer.

Pada tahun 2013 Shivani Jain dan A.L.N. Rao melakukan penelitian yang berjudul *Comparative Performance Analysis of Approximate String Matching*, dengan metode pengujian menggunakan parameter ukuran teks, panjang pola dan ukuran alphabet pada berbagai huruf dengan ukuran atau jenis teks yang berbeda yaitu *binary alphabet*, *alphabet of size 8*, *English alphabet* and *DNA alphabet*, yang memiliki karakteristik yang berbeda. Berdasarkan hasil empiris jelas bahwa algoritma *raita* lebih cepat dibandingkan algoritma *boyer moore*.. Dengan demikian algoritma *Raita* efisien dalam teori dan praktek untuk huruf kecil dan pola yang panjang [1].

Timo Raita dalam penelitiannya berjudul *Tuning the Boyer-Moore-Horspool String Searching Algorithm*, yaitu pengembangan yang bertumpu pada konsep algoritma *Boyer-Moore*. Didapatkan bahwa variansi baru dari *BMH* atau *Boyer Moore Horspool* mampu

meningkatkan performa BMH sebesar 21-27 persen. Pada pengujian algoritma ini, Timo Raita menggunakan teks sepanjang 29.550 karakter. Alfabet menggunakan ASCII dengan ukuran alfabet secara teori 128 dan ukuran aktualnya 85. Pola yang digunakan ialah pola hasil pengacakan dari teks dengan panjang 2-20. Pencarian dilakukan sebanyak 30 kali setiap masing-masing panjang pola [2].

Pada penelitian yang dilakukan oleh Syarief Ali Darwis pada tahun 2013 mengenai aplikasi pencarian data pegawai pada BPTP Kal-Sel menggunakan algoritma *Boyer-Moore* dari seluruh total waktu rata-rata pengujian yang dilakukan menghasilkan tingkat kecepatan 2.8150 ms yaitu 320 ms atau 1 % lebih lambat dari algoritma *binary search* dengan kecepatan 2.7830 ms, dengan 5 kali simulasi pada setiap kata kunci, dengan metode pengujian menggunakan teknik pengujian *Matchcase, Partcase, String, Date, Numerik*, dan berdasarkan variasi jumlah data 96 *record data*, 64 *record data* dan 32 *record data* [3].

Tulisan ini menggunakan data yang sama dengan metode yang berbeda dengan tujuan untuk mengetahui ketepatan data yang ditemukan dan kecepatan algoritma *Raita*, pada penelitian ini algoritma *Raita* akan diuji dengan cara membandingkan hasil pencarian dengan algoritma *Binary Search*.

2. Metode Penelitian

2.1. Algoritma Raita

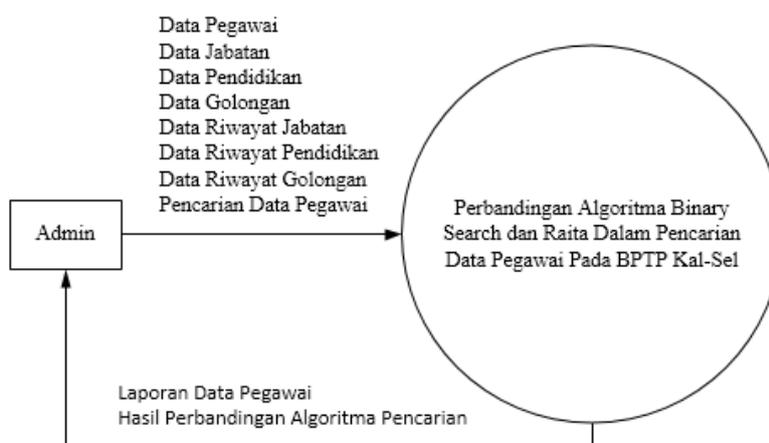
Algoritma Raita disajikan seperti berikut [4]:

1. Pertama membandingkan karakter pola terakhir, kemudian yang pertama dan akhirnya yang di tengah sebelum benar-benar membandingkan yang lain.
2. Melakukan pergeseran seperti algoritma horspool.
3. Fase preprocessing di $O(m + \sigma)$ waktu dan $O(\sigma)$ kompleksitas ruang.
4. Fase mencari di $O(m \times n)$ waktu kompleksitas.

2.2. Perancangan Penelitian

1. Diagram Konteks

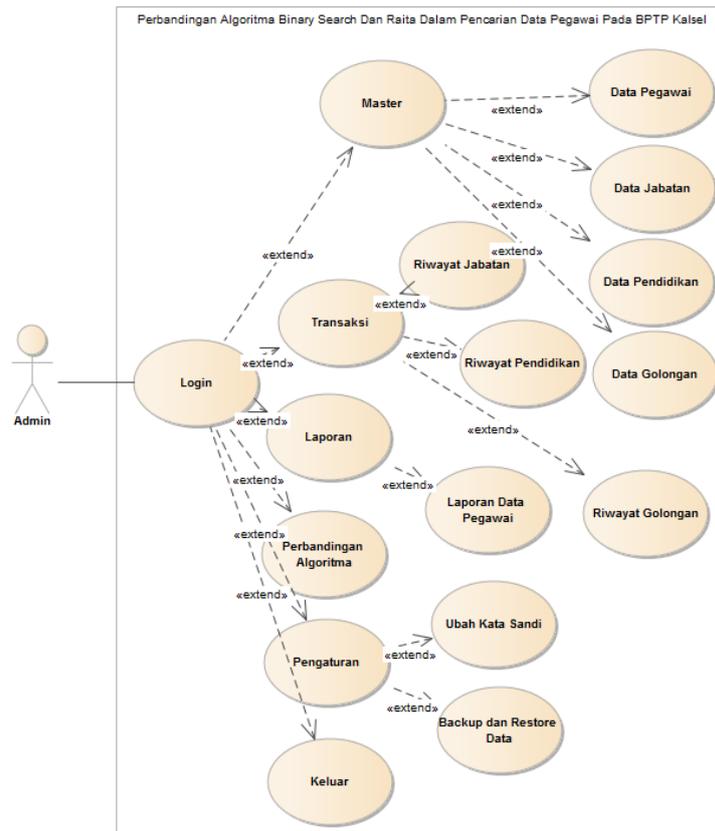
Diagram konteks adalah gambaran umum mengenai sistem prosedur yang terjadi antara sistem dan penggunanya. Diagram konteks dari sistem ini ditunjukkan pada gambar 1:



Gambar 1. Diagram Konteks

2. Diagram Konteks

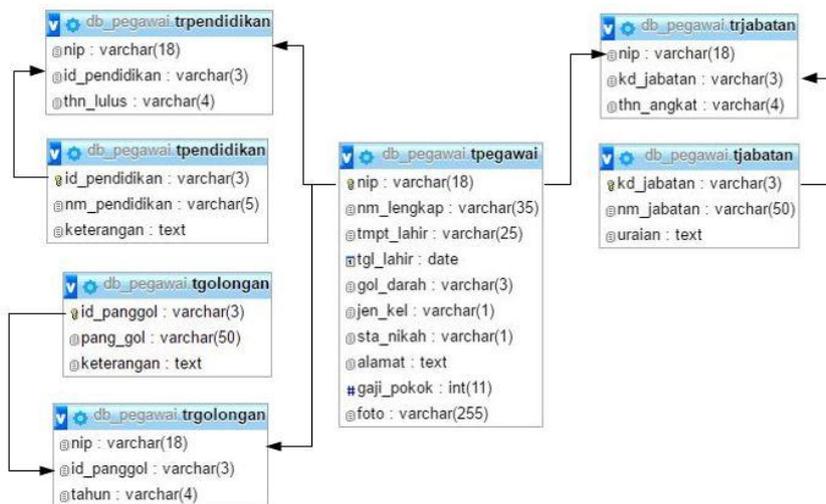
Use Case adalah konstruksi untuk mendeskripsikan bagaimana sistem terlihat dimata pengguna. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan yang disepakati antara pemakai dan pengembang. Gambar 2 menjelaskan tentang bagaimana interaksi admin dalam mengoperasikan aplikasi perbandingan pencarian data pegawai pada BPTP Kalsel.



Gambar 2. Use Case Diagram

3. Relasi Database

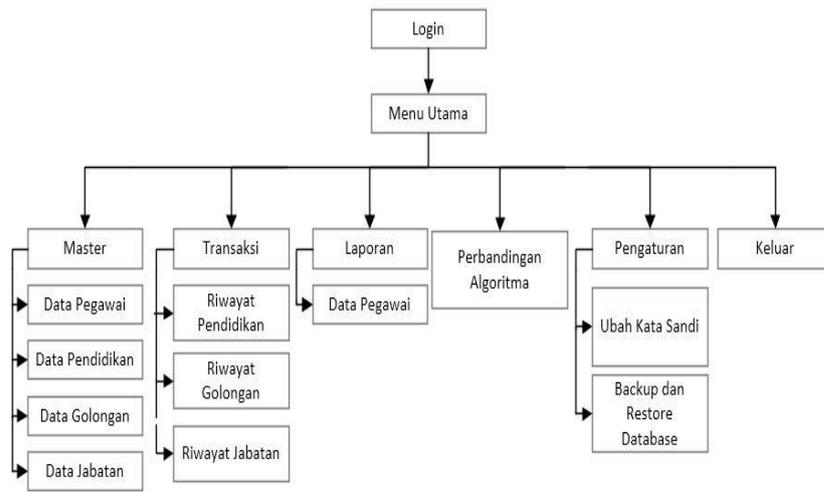
Pada sistem yang dibangun, table-table yang telah dirancang, akan direlasiikan berdasarkan key pada tabel-tabel. Relasi antar tabel dapat dilihat pada gambar berikut:



Gambar 3. Relasi Tabel

4. Desain Arsitektural Aplikasi

Pada sub bab ini digambarkan mengenai desain arsitektur (*site map*) sistem yang akan dibuat. Yang terdiri dari menu master, transaksi, laporan, perbandingan algoritma, pengaturan dan keluar beserta submenunya sebagai berikut:



Gambar 4. Desain Arsitektur Sistem

2.2. Teknik Analisa Data

Untuk menganalisa algoritma, tahap-tahap yang dilakukan adalah sebagai berikut :

- a) Sampel data dijadikan sebagai sumber data untuk pencarian. Sebagai contoh, admin mencari data dengan kata “HUSIN” dari data pegawai yang tersedia. Langkah-langkah algoritma *Raita* dalam pencarian data tersebut adalah sebagai berikut :

- 1) Preprocessing phase yaitu menentukan nilai table bmBC

T:	A	B	D	U	L		S	A	B	U	R
P:	H	U	S	I	N						
		4	3	2	1						

	H	I	S	U	*
bmBC	4	1	2	3	5

- 2) Mencocokkan kata kunci “HUSIN” dengan setiap data

Mencocokkan kata kunci dengan kata pada data pertama, yaitu mencocokkan “HUSIN” dengan “ABDUL SABUR”. Pertama mencocokkan karakter dari kanan atau terakhir terlebih dahulu yaitu karakter “N” dengan karakter yang sejajar yaitu karakter “L”, jika kedua karakter tidak cocok maka bergeser 5 langkah kekanan. Nilai pergeseran sesuai dengan yang ada pada tabel bmBC.

A B D U L S A B U R
 H U S I N

Kemudian mencocokkan karakter “N” dengan karakter yang sejajar berikutnya yaitu karakter “U”, jika kedua karakter tidak cocok maka bergeser 3 langkah kekanan.

A B D U L S A B U R
 H U S I N

Karena pergeseran melebihi panjang kata pada data pertama, maka pencarian berhenti dengan hasil tidak ada kecocokan pada data pertama.

A B D U L S A B U R
 H U S I N

Kemudian proses pencocokan dengan data selanjutnya, yaitu "NANANG HUSIN". Pertama mencocokkan karakter "N" dengan karakter yang sejajar yaitu karakter "N".

```

N A N A N G   H U S I N
H U S I N
    
```

Karena kedua huruf cocok maka, pencocokan dilanjutkan pada karakter paling kiri atau pertama yaitu karakter "H" dengan karakter yang sejajar yaitu karakter "N". jika kedua huruf tidak cocok maka geser 5 langkah kekanan.

```

N A N A N G   H U S I N
H U S I N
    
```

Kemudian pencocokan karakter "N" dengan karakter "S", karena kedua karakter tidak cocok maka geser kekanan 2 langkah.

```

N A N A N G   H U S I N
                H U S I N
    
```

Selanjutnya mencocokkan karakter "N" dengan karakter "N".

```

N A N A N G   H U S I N
                H U S I N
    
```

Karena kedua karakter cocok, maka pencocokan dilanjutkan pada karakter paling kiri yaitu karakter "H" dengan karakter "H".

```

N A N A N G   H U S I N
                H U S I N
    
```

Kedua karakter kembali cocok, pencocokan dilanjutkan pada karakter menengah yaitu karakter "S" dengan karakter "S".

```

N A N A N G   H U S I N
                H U S I N
    
```

Kedua karakter kembali cocok, pencocokan dilanjutkan pada karakter kedua dari kata kunci yaitu karakter "U".

```

N A N A N G   H U S I N
                H U S I N
    
```

Kedua karakter kembali cocok, pencocokan dilanjutkan pada karakter selanjutnya kearah kanan dari kata kunci yang belum dicocokkan yaitu karakter "I" dengan karakter "I".

```

N A N A N G   H U S I N
                H U S I N
    
```

Dapat dilihat diatas, karakter "H", "U", "S", "I", dan "N" sudah cocok pada data "NANANG HUSIN" dengan kata kunci "HUSIN". Kemudian proses dilanjutkan dan berhenti pada batas akhir data.

- b) Menghitung hasil data yang ditemukan dan menghitung kecepatan pencarian data
- c) Membandingkan kecepatan pencarian antara algoritma *Raita* dan algoritma *Binary Search*.

3. Hasil Dan Pembahasan

3.1. Hasil Pengujian Algoritma

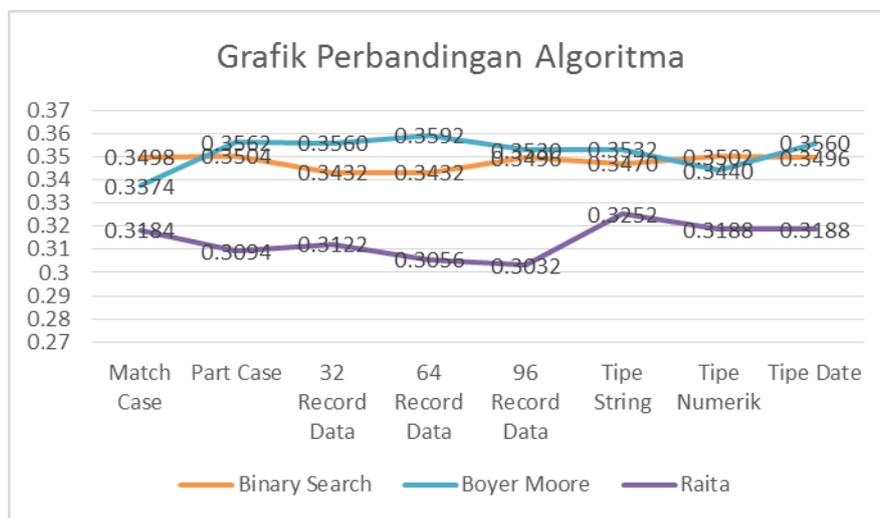
Pengukuran pada penelitian ini menggunakan pengukuran pretest dan posttest. Pretest adalah keadaan hasil pengujian pencarian data sebelum menggunakan algoritma Raita, yaitu menggunakan algoritma Binary Search dan Boyer Moore sedangkan posttest adalah hasil setelah dilakukan pencarian menggunakan algoritma Raita. Hal ini bertujuan untuk mengetahui beda hasil pengujian metode pencarian sebelumnya dengan hasil pengujian menggunakan algoritma *raita*.

Waktu rata-rata setiap teknik pencarian sebagai berikut.

Binary Search = $0.3498+0.3504+0.3432+0.3432+0.3496+0.347+0.3502+0.3496$
= 2.7830 ms.

Boyer Moore = $0.3374+0.3562+0.3560+0.3592+0.3530+0.3532+0.3440+0.3560$
= 2.8150 ms.

Raita = $0.3184+0.3094+0.3122+0.3056+0.3032+0.3252+0.3188+0.3188$
= 2.5116 ms.



Gambar 5. Grafik Perbandingan Algoritma

4. Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dalam hal ini membandingkan keakuratan berdasarkan kecepatan dan ketepatan pencarian antara algoritma Binary Search, boyer moore dan Raita, maka dapat disimpulkan bahwa dari 8 teknik pengujian yaitu match case, part case, 32 record data, 64 record data, 96 record data, tipe data string, tipe data numeric dan tipe data date dengan 5 kali simulasi dari setiap kata kunci pada teknik pengujian, algoritma raita lebih cepat 3% dari algoritma binary search dalam pencarian data dari keseluruhan data sampel yang diuji dengan total waktu 2.5116 ms dibandingkan penelitian terdahulu yang mana algoritma boyer moore dengan total waktu 2.8150 ms lebih lambat 1% dari algoritma binary search dengan total waktu 2.7830 ms. Hal tersebut diperoleh dengan menghitung total waktu rata-rata pada setiap kategori teknik pengujian kemudian dijumlahkan menjadi total waktu setiap algoritma pencarian.

Daftar Pustaka

- [1] Jain A., Rao A. (2013) "A Comparative Performance Analysis of Approximate String Matching," *International Journal of Innovative Technology and Exploring Engineering*, 3(5).pp.126-128
- [2] Raita T. (1992) "Tuning the Boyer–Moore–Horspool String Searching Algorithm," *SOFTWARE—PRACTICE AND EXPERIENCE*, 22. pp 883.
- [3] Darwis S.A. (2013) *Aplikasi Pencarian Data Pegawai Pada BPTP Kalsel*, Banjarbaru: STMIK BANJARBARU
- [4] Charras C., Lecroq T., (2001) "Handbook of Exact String Matching".

