

Prototipe Pencarian Berkas Kinerja Menggunakan Algoritma Knuth Morris Pratt (Studi Kasus pada Lembaga Amil Zakat)

Hanifah Permatasari*¹, Eko Purwanto², Triyono³

^{1,2}Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Duta Bangsa Surakarta. Jalan Bhayangkara No 55, Tipes, Serengan Surakarta, 57154
e-mail: ¹hanifah_permatasari@udb.ac.id, ²eko_purwanto@udb.ac.id

³Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Duta Bangsa Surakarta.
e-mail: ³triyono@udb.ac.id

Submitted Date: November 07th, 2023
Revised Date: January 23rd, 2024

Reviewed Date: January 23rd, 2024
Accepted Date: January 24th, 2024

Abstract

Operational activities at the Amil Zakat Institution (LAZ) produce various files or documents or reports or archives every year. The problem facing institutions is that the number of these reports will continue to increase over time. The management of these files has been carried out digitally using various types of information technology. The focus of information technology is not only being able to store files but also being able to find them again, so this research was conducted to optimize the performance of the search feature on file management information systems. This optimization is carried out by applying the Knuth Morris Pratt (KMP) Algorithm. The research stage is to design an algorithm for the system, build an application based on the design that has been carried out, prepare the data to be tested, and carry out testing. This research has resulted in a prototype LAZ file search. The results of testing this prototype is that the KMP Algorithm has no significant impact on search than the usual SQL Query on PHP. The test results show that the search time for files in all folders only increases by 0.095%, and the search time for files in one folder increases only by 0.007%.

Keywords: File Search; Amil Zakat Institutions; The Knuth Morris Pratt Algorithm

Abstrak

Kegiatan operasional pada Lembaga Amil Zakat (LAZ) menghasilkan berbagai berkas atau dokumen atau laporan atau arsip yang banyak setiap tahunnya. Permasalahan yang dihadapi lembaga adalah jumlah laporan-laporan ini akan terus bertambah seiring berjalannya waktu. Manajemen berkas-berkas tersebut telah dilakukan secara digital dengan menggunakan berbagai jenis teknologi informasi. Fokus teknologi informasi tidak hanya mampu menyimpan berkas namun juga mampu menemukannya kembali, sehingga penelitian ini dilakukan untuk mengoptimalkan kinerja fitur pencarian pada sistem informasi manajemen berkas. Optimisasi ini dilakukan dengan menerapkan Algoritma Knuth Morris Pratt (KMP). Tahap penelitiannya adalah merancang algoritma pada sistem, membangun aplikasi berdasarkan perancangan yang telah dilakukan, menyiapkan data yang akan diuji, dan melakukan pengujian. Penelitian ini telah menghasilkan prototipe pencarian berkas LAZ. Hasil uji terhadap prototipe ini adalah Algoritma KMP tidak berdampak signifikan dalam pencarian daripada SQL Query biasa pada PHP. Hasil uji menunjukkan hasil waktu pencarian berkas di seluruh folder hanya meningkat 0,095%, dan hasil waktu pencarian berkas di satu folder hanya meningkat 0,007%.

Kata Kunci: Pencarian Berkas; Lembaga Amil Zakat; Algoritma Knuth Morris Pratt



1 Pendahuluan

Lembaga Amil Zakat (LAZ) merupakan lembaga nirlaba yang bergerak di bidang filantropi. Lembaga ini milik yayasan swasta yang dikelola oleh organisasi masyarakat Islam, serta berperan dalam pengumpulan, pengelolaan, dan pendistribusian zakat di bawah izin Badan Amil Zakat Nasional (BAZNAS). Adapun izin operasional yang diperoleh LAZ dari BAZNAS tidak berlaku satu kali untuk seumur hidup, melainkan ada masa berlakunya (Fathony, 2018). BAZNAS akan melakukan audit kinerja dan keuangan dua kali dalam setahun, guna menjamin akuntabilitas kinerja LAZ. Apabila izin operasional LAZ telah selesai masa berlakunya, maka LAZ akan mengajukan kembali, dengan mengumpulkan beberapa persyaratan yang ditentukan oleh BAZNAS.

Misi BAZNAS dan LAZ adalah sama-sama berupaya untuk mengelola dana zakat dari masyarakat untuk didistribusikan sesuai dengan sasaran dan porsinya. Selain itu, BAZNAS dan LAZ juga menjalin kerjasama dengan beberapa mitra guna membuat penyaluran zakat menjadi lebih produktif. Artinya, masyarakat yang membutuhkan tidak hanya diberikan bantuan, akan tetapi juga diberdayakan sehingga perekonomian masyarakat bisa bertumbuh.

LAZ menjalankan misinya untuk memberdayakan zakat dengan memiliki beberapa program penghimpunan yang melahirkan konsekuensi adanya pelaporan kegiatan yang transparan dan akuntabel. Pelaporan ini ditujukan kepada beberapa pemangku kepentingan, termasuk untuk masyarakat yang berpartisipasi dalam aktivitas penghimpunan (Marliyati et al., 2017). Jumlah laporan-laporan ini akan terus bertambah seiring berjalannya waktu.

Pada sistem yang berjalan saat ini, masing-masing program penghimpunan akan menghasilkan laporan realisasi. Satu laporan realisasi akan membutuhkan beberapa data atau informasi referensi. Jumlah dokumen atau berkas atau arsip yang disimpan, dan didistribusikan akan terus bertambah seiring berjalannya waktu. Manajemen berkas atau arsip digital menggunakan sistem informasi sudah dilakukan oleh penelitian sebelumnya (Permatasari & Nofikasari, 2021). Akan tetapi, belum ada optimasi khusus pada fitur pencarian, mengingat salah fungsi dari manajemen

berkas digital adalah menemukan kembali (Iswandi et al., 2019).

Guna meningkatkan performa fitur pencarian, penelitian terdahulu telah mengimplementasikan algoritma pencarian berkas pada sistem informasi atau prototipe manajemen berkas digital. Algoritma ini diklaim memiliki pengaruh yang signifikan untuk mencari berkas jika dibandingkan dengan Query MySQL biasa. Beberapa algoritma yang pernah digunakan adalah *Knuth Morris Pratt*, *Brute Force*, *Sequential Searching*, *Turbo Booyer Moore*, *Binary Search*.

Penelitian ini bertujuan untuk mengembangkan prototipe pencarian berkas LAZ dengan menggunakan Algoritma *Knuth Morris Pratt* (*KMP*). Algoritma ini mencari pola susunan kata dari arah kiri ke kanan. Algoritma *KMP* telah digunakan pada penelitian terdahulu untuk mencari berkas digital. Algoritma *KMP* ini dinilai efisien karena meminimalkan jumlah perbandingan pola terhadap *string* input (Adline et al., 2022).

Penelitian pertama, Algoritma *KMP* diimplementasikan untuk pencarian berkas transaksi pada shipment ekspor dan impor (Khasanah, 2018). Penelitian kedua, Algoritma *KMP* diimplementasikan untuk pencarian berkas tugas akhir pada repository (Diana et al., 2022). Penelitian ketiga, Algoritma *KMP* diimplementasikan untuk pencarian seluruh berkas administratif pada Dinas Pariwisata (Aryasa et al., 2022). Penelitian keempat, Algoritma *KMP* diimplementasikan untuk pencarian kata pada kamus farmasi berbasis android (Ade Siti Nur zainab, 2020). Penelitian kelima, Algoritma *KMP* diimplementasikan untuk pencarian surat di lingkungan sekolah (Ilham & Mirza, 2020). Penelitian keenam, Algoritma *KMP* diimplementasikan untuk pencarian dalam *search engine* (Priyatna, 2023)

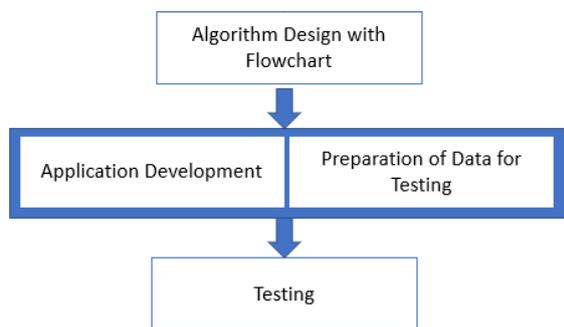
Sementara itu, belum ada penelitian sama sekali di lingkungan LAZ, padahal pertumbuhan berkas atau arsip di lingkungan ini juga tinggi. Mengingat banyaknya program penghimpunan dan pemberdayaan zakat yang harus dipertanggungjawabkan setiap tahunnya.

Adapun pembuatan prototipe pencarian berkas dengan algoritma *KMP* pada penelitian ini akan melalui beberapa tahap, di antaranya adalah merancang algoritma pada sistem, membangun aplikasi berdasarkan perancangan yang telah dilakukan, menyiapkan data yang akan diuji, dan

melakukan pengujian. Penelitian ini juga menemukan seberapa efektivitas dari algoritma KMP yang diterapkan pada sistem pencarian berkas digital.

2 Metode Penelitian

Penelitian ini mengambil beberapa contoh laporan dalam kegiatan distribusi zakat. Data tersebut akan dimasukkan pada prototipe yang sedang dikembangkan untuk diuji. Tahap pengembangan prototipe dapat dilihat pada Gambar 2.



Gambar 2. Tahap Penelitian

Tahap pertama dimulai dengan melakukan perancangan algoritma dengan flowchart. Selanjutnya, aplikasi akan dikembangkan, bersamaan dengan menyiapkan data untuk menguji algoritma pencarian. Langkah terakhir adalah melakukan pengujian, baik terhadap aplikasi maupun algoritma.

2.1 Perancangan Algoritma

Cara kerja algoritma KMP terdapat pada gambar berikut ini:

01	Enter the word to be searched for, for example P = Pattern or word order pattern that will be searched on T = Text or document title
02	The Knuth Morris Pratt algorithm begins to match the pattern or word order pattern used as an example at the beginning of the text
03	Starting from left to right, this algorithm will match each character pattern or word pattern that is used as an example with the corresponding text character from left to right
a	The pattern or pattern of the character arrangement used as an example and the text being compared does not match or does not match
b	All patterns of word order or characters in the pattern used as examples are appropriate or suitable, then the KMP algorithm will notify the finding of the matching pattern.
04	the KMP algorithm will shift the word order pattern component or pattern example based on the text table, then repeat step number 2 until the word order pattern used as an example is at the end of the text

Gambar 3. Cara Kerja Algoritma KMP

Berikut adalah ilustrasi cara kerja Algoritma KMP di dalam Prototipe Pencarian Berkas Kinerja Lembaga Amil Zakat. Data telah dibagi menjadi dua bagian sesuai dengan karakteristik pencarian.

2.2 Data Pertama

Pada data pertama, akan dilakukan pencocokan antara *string* dengan *pattern* yang sama, sesuai dengan urutannya. Adapun kata yang kita cocokkan adalah PANITIA, terdiri dari tujuh lebar *string* dan tujuh lebar *pattern*. Langkahnya adalah sebagai berikut:

- a) Pencocokan *String* [1] dengan *Pattern* [1]

<i>String</i>	p	a	n	i	t	i	a
<i>Pattern</i>	p	a	n	i	t	i	a

Ketika informasi antara *string* dan *pattern* cocok, maka algoritma KMP akan menyimpannya. Algoritma akan berjalan mencocokkan *string* dan *pattern* selanjutnya. Contohnya, diketahui bahwa ada kecocokan antara *String*[1] dan *Pattern*[1], sehingga informasi tersebut akan disimpan. *Pattern* tidak bergeser dan algoritma akan berjalan mencocokkan *String*[2] dengan *Pattern*[2]. Hal ini juga bisa dilihat pada poin (b) sampai (g) di bawah ini.

b) Pencocokan *String* [2] dengan *Pattern* [2]

```
String  p a n i t i a
Pattern p a n i t i a
```

c) Pencocokan *String* [3] dengan *Pattern* [3]

```
String  p a n i t i a
Pattern p a n i t i a
```

d) Pencocokan *String* [4] dengan *Pattern* [4]

```
String  p a n i t i a
Pattern p a n i t i a
```

e) Pencocokan *String* [5] dengan *Pattern* [5]

```
String  p a n i t i a
Pattern p a n i t i a
```

f) Pencocokan *String* [6] dengan *Pattern* [6]

```
String  p a n i t i a
Pattern p a n i t i a
```

g) Pencocokan *String* [7] dengan *Pattern* [7]

```
String  p a n i t i a
Pattern p a n i t i a
```

2.3 Data kedua

Pada data kedua, terdapat *string* dan *pattern* yang berbeda susunan karakter dan ukurannya. Kata yang kita cocokkan adalah PEMBERI INFAQ, sementara *pattern* yang ada adalah PANITIA. Kedua kata tersebut memiliki lebar yang berbeda, sehingga langkah pencocokannya adalah sebagai berikut:

a) Pencocokan *String* [1] dengan *Pattern* [1]

```
String  p e m b e r i i n f a q
Pattern p a n i t i a
```

Informasi *String*[1] dan *Pattern*[1] akan disimpan karena telah cocok. *Pattern* tidak bergeser dan berjalan mencocokkan *String*[2] dengan *Pattern*[2].

b) Pencocokan *String*[2] dengan *Pattern*[2]

```
String  p e m b e r i i n f a q
Pattern p a n i t i a
```

Apabila *string* dengan *pattern* tidak cocok, maka *pattern* akan bergeser satu posisi ke kanan. Contohnya adalah poin (b), diketahui bahwa *String*[2] tidak cocok dengan *Pattern*[2]. Maka dari

itu *pattern* akan bergeser satu posisi ke kanan. Hal ini juga berlaku pada poin (c) sampai (j).

c) Pencocokan *String*[2] dengan *Pattern*[1]

```
String  p e m b e r i i n f a q
Pattern  p a n i t i a
```

d) Pencocokan *String*[3] dengan *Pattern*[1]

```
String  p e m b e r i i n f a q
Pattern  p a n i t i a
```

e) Pencocokan *String*[4] dengan *Pattern*[1]

```
String  p e m b e r i i n f a q
Pattern  p a n i t i a
```

f) Pencocokan *String*[5] dengan *Pattern*[1]

```
String  p e m b e r i i n f a q
Pattern  p a n i t i a
```

g) Pencocokan *String*[6] dengan *Pattern*[1]

```
String  p e m b e r i i n f a q
Pattern  p a n i t i a
```

h) Pencocokan *String*[7] dengan *Pattern*[1]

```
String  p e m b e r i i n f a q
Pattern  p a n i t i a
```

Algoritma KMP akan berhenti dikarenakan 7 karakter pada *pattern* tidak ada yang sesuai dengan data *string*, sehingga *string* pemberi infaq tidak dapat ditampilkan pada tabel data *file* berkas dan akan melakukan pencocokan pada data selanjutnya.

3 Hasil dan Pembahasan

Pada bagian ini terdiri dari dua pembahasan, yaitu (1) Pengembangan Prototipe dan Penyiapan Data Uji, dan (2) Pengujian. Pada bagian pengembangan prototipe dan penyiapan data uji, akan dijelaskan struktur *coding* dari prototipe berikut dengan data yang hendak diuji. Adapun data yang dimaksud adalah data contoh berkas yang ada pada Lembaga Amil Zakat. Sementara

pada bagian pengujian, akan dijelaskan tentang bagaimana efektivitas yang dihasilkan Algoritma KMP terhadap prototipe yang telah dikembangkan.

Pengembangan Prototipe dan Penyiapan Data Uji

Pada bagian ini akan ditunjukkan struktur *coding* Algoritma KMP pada prototipe yang dikembangkan, beserta dengan penjelasannya. Algoritma ini akan digunakan untuk mencari berkas pada antarmuka yang dioperasikan oleh pengguna. Berikut adalah penjelasannya:

a. Coding Algoritma KMP pada Prototipe Computing Longest Prefix Suffix (LPS)

```
List computeLPS(String pattern) {  
    List lps = List.filled(pattern.length, 0);  
    lps[0] = 0;
```

LPS selalu 0 karena sufiks harus selalu dimulai minimal dari indeks 1 (karakter kedua)

```
int panjangPattern = pattern.length;  
int j = 0;  
int i = 1;
```

```
while (i < panjangPattern) {  
    if (pattern[j] == pattern[i]) {
```

Jika karakter cocok maka akan menambah nilai j dan i

```
lps[i] = j + 1;  
i++;  
j++;  
} else if (j > 0) {  
    j = lps[j - 1];  
} else {
```

tidak ada yang cocok, lps pertama akan direset ke 0 dan i ditambah 1

```
lps[i] = 0;  
i++;  
}  
}  
logger.d('lps $lps');  
return lps;  
}
```

KMP (Knuth-Morris-Pratt)

```
List kmp(String data, String pattern) {  
    List foundIndexes = <int>[];  
    int panjangData = data.length;  
    int panjangPattern = pattern.length;  
    int i = 0;  
    int j = 0;  
    List lps = computeLPS(pattern);  
    while (i < panjangData) {
```

selama i kurang dari panjang data akan selalu diulang

```
if (pattern[j] == data[i]) {
```

jika karakter cocok maka akan menambah nilai j dan i

```
i++;  
j++;  
}
```

jika j sama dengan panjang *pattern* akan memasukkan data ke index yang ditemukan

```
if (j == panjangPattern) {  
    foundIndexes.add(i -  
    panjangPattern);  
    logger.d(j);  
    j = lps[j - 1];  
    i += panjangPattern - 1;  
} else if (i < panjangData &&  
    pattern[j] != data[i])
```

jika i kurang dari panjang data dan pola tidak sama dengan data[i]

```
{  
    if (j != 0) {
```

jika j tidak sama dengan 0 maka j = diisi dengan data pada lps index dikurangi 1

```
j = lps[j - 1];  
} else {
```

selain itu i ditambahkan dirinya sendiri tambah 1

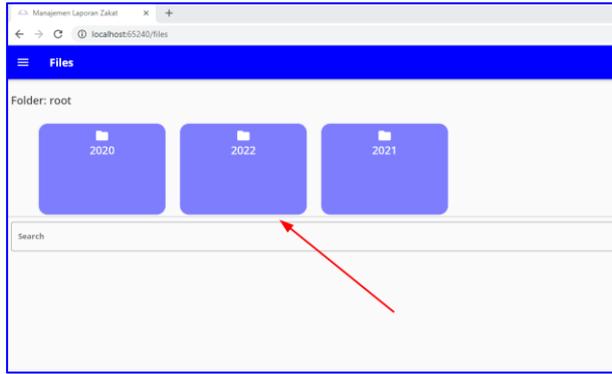
```
i = i + 1;
```

setelah selesai mengembalikan data yang ketemu

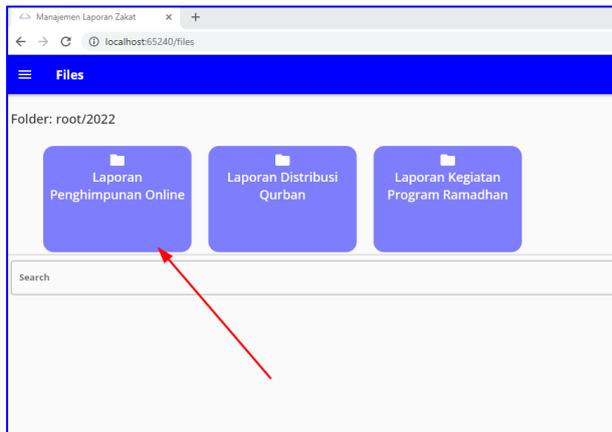
```
return foundIndexes;  
}
```

b. Data Uji dan Hasil Antarmuka

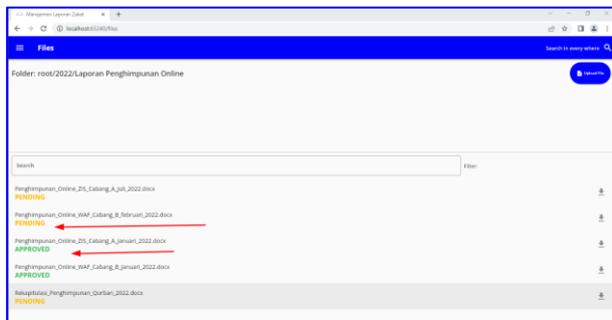
Prototipe Pencarian Berkas Kinerja LAZ dikembangkan berbasis web. Adapun data yang diujikan dalam prototipe ini adalah data-data foto dan laporan kinerja LAZ. Data tersebut berbentuk DOCX dan JPG/PNG. Ada sejumlah data yang diisikan pada setiap tahun, karena prototipe dikembangkan untuk memiliki dua fitur pencarian. Pertama, pencarian pada folder tertentu. Kedua, pencarian di seluruh folder yang ada. Berikut adalah hasil dari pengembangan prototipe:



Gambar 4. Halaman Root Berkas Per Tahun



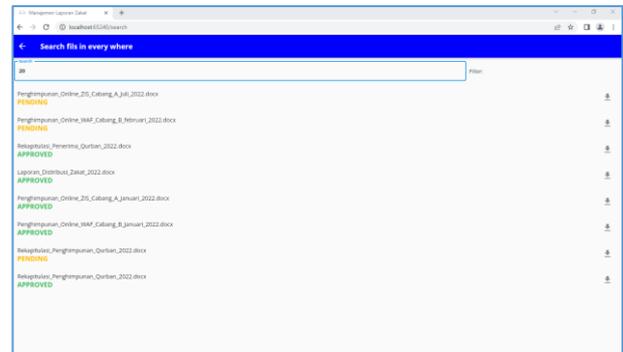
Gambar 5. Halaman Menampilkan Berkas di Tahun X



Gambar 6. Halaman Menampilkan Berkas di Tahun X



Gambar 7. Hasil Pencarian Berkas di dalam Folder



Gambar 8. Hasil Pencarian Berkas di seluruh Folder

Pengujian

Prototipe dan algoritma diuji dengan dua kasus, yaitu mencari *file* di satu folder, dan *file* di seluruh folder. Hasilnya bisa dilihat ada Tabel 1 dan Tabel 2.

Tabel 1. Hasil Waktu Pencarian Berkas di Seluruh Folder

No	Kata Pencarian	KMP (Micro second)	Tanpa KMP
1	Foto	122400	122399
2	Online	178799	178759
3	Laporan	196801	196801
4	Zakat	144400	144499
5	2022	310801	310500
6	Januari	130400	129990
7	Cabang	214000	214088
8	Juli	122000	123990
9	Distri	112900	112955
10	Acara	115000	115088
Rata-Rata		164750	164907
		Peningkatan	0,095%

Tabel 2. Hasil Waktu Pencarian Berkas di Satu Folder

No	Kata Pencarian	KMP (Micro second)	Tanpa KMP
1	Maret	88400	88410
2	Himpunan	118801	118809
3	Cabang	140600	140605
4	Online	105000	105004
5	Juli	73700	73709
Rata-Rata		105300,2	105307,4
		Peningkatan	0,007%

4 Kesimpulan

Prototipe pencarian berkas kinerja LAZ ini telah dikembangkan berbasis website, dan menerapkan Algoritma Knuth Morris Pratt untuk mempersingkat waktu pencarian berkas digital. Aplikasi ini memungkinkan pengguna mencari berkas dalam satu folder, dan mencari berkas dalam seluruh folder yang ada di dalam sistem. Hasilnya adalah terdapat perbedaan waktu respon sistem antara menggunakan Algoritma KMP dengan SQL Query biasa pada PHP.

Hasil uji menunjukkan hasil waktu pencarian berkas di seluruh folder hanya meningkat 0,095%, dan hasil waktu pencarian berkas di satu folder hanya meningkat 0,007%. Artinya, optimasi fitur pencarian menggunakan Algoritma KMP tidak terlalu berdampak signifikan dalam pencarian daripada SQL Query biasa pada PHP.

Penelitian selanjutnya dapat mencoba penerapan Algoritma KMP pada bahasa pemrograman yang lain, atau menerapkan algoritma pencarian yang lain pada bahasa pemrograman yang sama. Hasilnya dapat digunakan sebagai perbandingan, sehingga mampu memberikan rujukan bagi peneliti sistem informasi manajemen berkas atau arsip, dalam mengoptimasi fitur pencarian.

Referensi

- Ade Siti Nur zainab. (2020). Perbandingan Algoritma Horspool Dan Knuth Morris Pratt Pada Aplikasi Kamus Farmasi Berbasis Android. *Semantik Uho*, 6(1), 21–30. <http://ojs.uho.ac.id/index.php/semantik/article/view/10222/pdf>
- Adline, F., Tobing, T., Chandra, A., & Nainggolan, R. (2022). Penerapan Algoritma Knuth Morris Pratt (Kmp) Pada Pencarian Data Di Sql Like Operators. *Jurnal Widya, Vol.3 No.1*(April), 50–58. <https://jurnal.amikwidyaloka.ac.id/index.php/aw1>
- Aryasa, K., Likliwatil, R. D., Yosep, & Prierendi, R. (2022). Implementasi Algoritma Knuth Morris Pratt Dalam Pencarian Berkas Berbasis Web (Studi Kasus: Dinas Pariwisata Kota Makassar). *Jurnal Sistem Informasi Dan Teknologi Informasi*, 11(1), 1–12. <https://doi.org/https://doi.org/10.36774/jusiti.v11i1.906>
- Diana, Amri, & Husaini. (2022). Pencarian Kata Dalam Konten Tugas Akhir Dengan Implementasi Algoritma Knuth Morris Pratt (KMP). *Jurnal Teknologi Rekayasa Informasi Dan Komputer*, 5(2), 3–6. <https://ejournal.pnl.ac.id/TRIK/article/viewFile/4789/3456>
- Fathony, A. (2018). Optimizing the Role and Function of the Amil Zakat Institution in Carrying Out Social Functions. *Journal Hakam*, 02(01), 1–32.
- Ilham, M., & Mirza, A. H. (2020). Pengarsipan Dokumen Pada Sma Plus Negeri 17 Palembang. *Bina Darma Conference on Computer Science*. <https://conference.binadarma.ac.id/index.php/B-DCCS/article/view/1526/772>
- Iswandi, N., Nazifah, N. A., Khotimah, H., Anggraini, M., & Okshi, J. (2019). Sistem Manajemen Arsip di MTS Aulia Cendikia Palembang. *Diplomatika: Jurnal Kearsipan Terapan*, 2(2), 65-74. <https://doi.org/10.22146/diplomatika.42038>
- Khasanah, N. (2018). Penerapan Algoritma Knuth Morris Pratt Pada Aplikasi Pencarian Berkas Shipment Berbasis Web (Studi Kasus di PT YEC Semarang). *E-Bisnis*, 11, 14–22. <https://journal.stekom.ac.id/index.php/E-Bisnis/article/view/93>
- Marliyati, Nikmatuniyah, & A, L. M. (2017). Effects of Accounting Information Quality , Accountability , and Transparency on Zakat Acceptance. *Mimbar*, 33(1), 62–73.
- Permatasari, H., & Nofikasari, I. (2021). Konsep Desain Sistem Informasi Manajemen Berkas Terpusat di Lembaga Amil Zakat Menggunakan Perspektif Nirlaba. *JUSIFO (Jurnal Sistem Informasi)*, 7(2), 65–80. <https://doi.org/https://doi.org/10.19109/jusifo.v7i2.9390>
- Priyatna, R. D. (2023). Implementasi Algoritma Levenstein Distance Dan Algoritma Knutt Morris Pratt Dalam Fitur Word Completion Pada Search Engine. *Nusantara Journal of Multidisciplinary Science*, 1(1), 145–156. <https://ejournal.pnl.ac.id/TRIK/article/viewFile/4789/3456>