

# Implementasi Algoritma Knuth-Morris-Pratt dalam Fitur Pencarian Pengarsipan Dokumen Berbasis Web (Studi Kasus: Vitech Asia Group, Jakarta)

Aji Gunawan<sup>1\*</sup>, Dimas Abisono Punkastyo<sup>1</sup>

<sup>1</sup>Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspipetek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan, Banten 15310, Indonesia

Email: <sup>1\*</sup>[ajig247@gmail.com](mailto:ajig247@gmail.com), <sup>2</sup>[dosen00675@unpam.ac.id](mailto:dosen00675@unpam.ac.id)

(\* : coressponding author)

**Abstrak**—Proses pencarian suatu dokumen arsip pada divisi administrasi di Vitech Asia Group masih menggunakan sistem sederhana dan manual yaitu setelah dokumen dicetak untuk selanjutnya dikelola sesuai dengan prosedur pengendalian dokumen yang ada, dokumen arsip disimpan di dalam lemari atau tempat pengarsipan. Dokumen yang dikelola akan terus bertambah seiring berjalannya waktu sehingga menyimpan dokumen secara manual tentu akan menyulitkan staff admin ketika mencari dokumen yang sedang dibutuhkan sehingga membuat pekerjaan menjadi tidak efektif dan efisien. Agar kegiatan operasional dapat berjalan dengan efektif, maka dibutuhkan sistem pencarian arsip yang terkomputerisasi untuk mengintegrasikan data-data arsip pada satu *database* yang terpadu agar pencarian suatu arsip dokumen dapat dilakukan dengan lebih efektif, cepat, tepat dan efisien. Pada penelitian ini akan mengimplementasikan algoritma *knuth-morris-pratt* pada perancangan sistem pencarian arsip yang berbasis web dengan menggunakan model pengembangan sistem yaitu model *waterfall*. Hasil dari penelitian ini berupa sistem pencarian arsip, dengan adanya sistem ini maka aktivitas pencarian dokumen menjadi lebih efektif, cepat, tepat dan efisien, di mana hal ini dibuktikan dengan hasil wawancara user, yang sebelumnya memerlukan waktu lebih dari 5 menit untuk mencari suatu dokumen arsip menjadi  $\pm 1$  menit dengan menggunakan sistem pencarian yang telah dibangun.

**Kata Kunci:** Sistem Pencarian Arsip, *Knuth-Morris-Pratt*, Web, *Waterfall*

**Abstract**—*The process of searching for an archival document in the administrative division at Vitech Asia Group still uses a simple and manual system, namely after the document is printed to be further managed in accordance with existing document control procedures, the archival document is stored in a cabinet or filing place. The documents managed will continue to grow over time so that storing documents manually will certainly make it difficult for admin staff when looking for documents that are needed, making work ineffective and efficient. In order for operational activities to run effectively, a computerized archive search system is needed to integrate archival data in one integrated database so that the search for a document archive can be carried out more effectively, quickly, precisely and efficiently. In this study, we will implement the knuth-morris-pratt algorithm in the design of a web-based archive search system using a system development model, namely the waterfall model. The results of this study are in the form of an archive search system, with this system, document search activities become more effective, fast, precise and efficient, where this is evidenced by the results of user interviews, which previously took more than 5 minutes to search for an archive document to be  $\pm 1$  minute using a search system that has been built.*

**Keywords:** Archive Search System, *Knuth-Morris-Pratt*, Web, *Waterfall*

## 1. PENDAHULUAN

PT Vitech Asia (Vitech Asia Group/VTA Group) merupakan perusahaan *IT System Integrator* yang bergerak di bidang Infrastruktur Teknologi Komunikasi, *IT Service* dan *Solution Provider* yang dibentuk pada tahun 2002. Dengan berjalannya proses bisnis dalam suatu organisasi, pengarsipan dokumen merupakan salah satu sumber informasi yang paling berpengaruh keberadaannya. Arsip adalah kumpulan data yang disimpan sebagai dokumen atau acuan apabila suatu saat data tersebut akan dipergunakan kembali, secara fisik arsip mengandung unsur-unsur struktur, isi dan konten dan dalam perkembangan zaman media penyimpanan arsip berubah dari objek yang bersifat fisik menjadi data dalam bentuk *software virtual* yang dapat dijangkau secara lebih mudah (Firdaus & Irfan, 2020:45).

Proses pengarsipan di VTA Group sendiri masih berjalan secara manual di mana setelah dokumen dicetak untuk selanjutnya dikelola sesuai dengan prosedur pengendalian dokumen yang ada, dokumen arsip disimpan di dalam lemari atau tempat pengarsipan. Dokumen yang dikelola akan terus bertambah seiring berjalannya waktu sehingga menyimpan dokumen secara manual tentu

akan menyulitkan staff admin ketika mencari dokumen yang sedang dibutuhkan sehingga membuat pekerjaan menjadi tidak efektif dan efisien. Selain itu penyimpanan dokumen secara manual tanpa adanya salinan secara elektronik memiliki resiko seperti kehilangan dan kerusakan dokumen.

Agar kegiatan operasional dapat berjalan dengan efektif, maka dibutuhkan sistem pencarian arsip yang terkomputerisasi untuk mengintegrasikan data-data arsip pada satu *database* yang terpadu, sehingga pengarsipan dokumen dapat dikelola dengan baik dan pada fitur pencarian arsip dokumen dalam sistem pencarian arsip menggunakan algoritma *string matching*. Algoritma *string matching* merupakan algoritma pencocokan *string*. Sifat algoritma *string matching* adalah mencari sebuah *string* yang terdiri dari beberapa karakter yang biasa disebut *pattern* dalam sejumlah besar *text* (Matondang, 2018:102). Terdapat beberapa algoritma pada *string matching* ini diantaranya *Brute Force*, *Winnowing*, *Knuth-Morris-Pratt*, *Booyer-Moore* dan sebagainya. Pada penelitian ini peneliti menerapkan algoritma *Knuth-Morris-Pratt* dalam fitur pencarian pengarsipan dokumen. Algoritma *Knuth-Morris-Pratt* merupakan algoritma pencarian *string* dengan melakukan perbandingan karakter mulai dari kiri ke kanan (Nursobah & Pahrudin, 2019:112). Algoritma *KMP* memiliki keunggulan pencarian kecocokan pada file yang berukuran besar (Sa'diah, 2017:115).

Pada penelitian ini akan mengimplementasikan algoritma *knuth-morris-pratt* pada perancangan sistem pencarian arsip yang berbasis web dengan menggunakan model pengembangan sistem yaitu model *waterfall* karena dengan model ini akan memudahkan dalam tahap pengembangan seperti yang dilakukan secara bertahap. Model *waterfall* adalah pengembangan atas perangkat lunak yang menitikberatkan melalui tahap-tahap yang sistematis dan secara berurutan. Analisis ini diawali dari proses pengumpulan kebutuhan, translasi kebutuhan perangkat lunak dalam tahap rancangan, pengkodean ke dalam tahap program serta *logic* dan fungsional pada tahap pengujian yang melalui serangkaian untuk mengidentifikasi kebutuhan pengguna atau konsumen tersebut pada tahap awal (Rofiq, Ferdananto & Nurjaya, 2021:19).

## 2. METODOLOGI PENELITIAN

### 2.1 Metode Pengumpulan Data

Metode pengumpulan data merupakan langkah yang paling strategis dalam penelitian, karena tujuan utama dari penelitian adalah mendapatkan data (Sugiyono, 2017:224). Dalam tahap pengumpulan data penulis menggunakan penelitian

#### a. Penelitian Lapangan (*Filed Research*).

Tahap-tahap yang digunakan yaitu wawancara yaitu dengan melakukan tanya-jawab secara langsung dengan mengajukan pertanyaan-pertanyaan kepada pihak-pihak yang terkait, seperti staff admin dengan tujuan untuk memperoleh data atau informasi yang berhubungan dengan masalah yang diteliti. Dan observasi yaitu pengamatan yang dilakukan oleh peneliti secara langsung terhadap aktivitas kerja di lingkungan kerja Vitech Asia Group.

#### b. Studi Kepustakaan

Studi Pustaka (*Library Research*), dilakukan untuk memperoleh data sekunder dengan melakukan penelaahan teori-teori yang berkaitan dengan topik penelitian yang berasal dari sumber-sumber penelitian kepustakaan yaitu buku, jurnal, hasil-hasil penelitian terdahulu yang telah dipublikasikan yang sesuai dengan topik penelitian

### 2.2 Metode Pengembangan Sistem

Dalam merancang sistem pencarian arsip ini penulis menggunakan *Sistem Development Live Cycle (SDLC)* dengan model *waterfall*. Di bawah ini akan dijelaskan langkah-langkah pengembangan *software* dengan menggunakan model *waterfall*:

#### a. Requirement

Pada proses ini penelitian dilakukan secara insentif untuk mengetahui kebutuhan perangkat lunak seperti apa yang dibutuhkan oleh user.

#### b. System and Software Design

Pada titik ini spesifikasi sistem permintaan pengguna dari *requirment analysis* penulis kemudian mengimplementasikannya dalam bentuk pengembangan desain berupa *Unifed Modeling Language (UML)* sedangkan pada perancangan basis datanya

menggunakan *Entity Relationship Diagram (ERD)*. UML merupakan kumpulan diagram-diagram yang sudah memiliki standar untuk membangun perangkat lunak berbasis objek” (Sulianta, 2017) sedangkan *ERD* adalah sebuah diagram yang menggambarkan setiap entitas yang terkait pada suatu sistem dimana pada *Physical Data Model*, setiap komponen yang ada pada *ERD* akan menjadi objek-objek yang disebut tabel, atribut, *data type*, *relationship*, *primary key*, *foreign key* dan objek lain (Simanjuntak dkk, 2017:15).

**c. Implementation**

Pada tahap ini, rencana aplikasi sistem diimplementasikan ke dalam bentuk program dengan menggunakan bahasa pemrograman *PHP*, *Framework Laravel* serta algoritma pencarian yang digunakan yaitu algoritma *Knuth-Morris-Pratt*. *PHP* adalah bahasa pemrograman yang digunakan bersamaan dengan *HTML* dimana bahasa tersebut dijalankan di sisi *web browser* (Nurmalasari, Anna & Arissusandi, 2019:10). Sedangkan *framework* adalah kumpulan dari fungsi-fungsi/prosedur-prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau *class* dari awal (Mualim & Putra, 2017:35). Algoritma *knuth-morris-pratt* merupakan algoritma pencarian *string* dengan melakukan perbandingan karakter mulai dari karakter paling kanan dari *string* yang dicari (Nursobah & Pahrudin, 2019:112),

**d. Integration and System Testing**

Dalam tahap ini semua urutan program menjadi satu kesatuan kemudian program diuji dan diperiksa secara menyeluruh untuk mengetahui apakah masih terdapat *bug* atau kesalahan pada sistem, pengujian yang dilakukan menggunakan *blackbox testing*. Metode ujicoba *Black Box* memfokuskan pada keperluan fungsional dari *software*, di mana Ujicoba *Black Box* berusaha untuk menemukan kesalahan dalam beberapa kategori diantaranya fungsi-fungsi yang salah atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses *database* eksternal, kesalahan performa dan kesalahan inisialisasi dan terminasi (Hidayat, 2017:93).

### 3. ANALISA DAN PEMBAHASAN

#### 3.1 Analisa

Secara sistematis langkah-langkah yang dilakukan algoritma *Knuth-Morris-Pratt* pada saat pencocokan *string* adalah sebagai berikut (Ramadhani, 2017:26):

- a. Masukkan *query* kata yang akan dicari. Dengan permissalan  $P = Pattern$  atau pola susunan kata yang dijadikan sebagai contoh atau pola teks yang akan dicari  $T = Teks$ .
- b. Algoritma *KMP* mulai mencocokkan *pattern* atau pola susunan kata yang dijadikan sebagai contoh pada awal teks.
- c. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter, *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
  1. Karakter di *pattern* dan di teks yang dibandingkan tidak cocok (*mismatch*).
  2. Karakter di *pattern* dan di teks cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
- d. Algoritma kemudian menggeser *pattern* atau pola susunan kata yang dijadikan sebagai contoh berdasarkan tabel *next*, lalu mengulangi langkah kedua sampai *pattern* atau pola susunan kata yang dijadikan sebagai contoh berada di ujung teks.

Perhitungan manual pada algoritma *Knuth-Morris-Pratt* digunakan untuk mendapatkan gambaran kasar tentang cara kerja algoritma ini. Data pada perhitungan manual ini diambil dari salah satu data nyata yang ada pada *database* terlihat seperti pada table.

**Tabel 1.** Pencocokan Pattern Dengan String Pada Index Pertama

p	e	t	u	n	j	u	k
t	u	n	j	u	k		

Langkah I: *Pattern* tidak cocok dengan *string*. Maka *pattern* bergeser satu posisi ke kanan.

**Tabel 2.** Pencocokan Pattern Dengan String Pada Index Kedua

p	e	t	u	N	j	u	k
	t	u	n	J	u	k	

Langkah II: *Pattern* tidak cocok dengan *string*. Maka *pattern* bergeser satu posisi ke kanan.

**Tabel 3.** Pencocokan Pattern Dengan String Pada Index Ketiga

p	e	t	u	N	j	u	k
		t	u	N	j	u	k

Langkah III: *Pattern* cocok dengan *string*. Karena ada kecocokan, maka algoritma *Knuth-Morris-Pratt* menyimpan informasi ini dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*.

**Tabel 4.** Pencocokan Pattern Dengan String Pada Index Keempat

p	E	T	u	N	j	u	k
		T	u	N	j	u	k

Langkah IV: *Pattern* cocok dengan *string*. Karena ada kecocokan, maka algoritma *Knuth-Morris-Pratt* menyimpan informasi ini dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*.

**Tabel 5.** Pencocokan Pattern Dengan String Pada Index Kelima

p	E	t	u	n	j	U	k
		t	u	n	j	U	k

Langkah V: *Pattern* cocok dengan *string*. Karena ada kecocokan, maka algoritma *Knuth-Morris-Pratt* menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*.

**Tabel 6.** Pencocokan Pattern Dengan String Pada Index Keenam

p	e	T	u	n	j	u	k
		t	u	n	j	u	k

Langkah VI: *Pattern* cocok dengan *string*. Karena ada kecocokan, maka algoritma *Knuth-Morris-Pratt* menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*.

**Tabel 7.** Pencocokan Pattern Dengan String Pada Index Ketujuh

p	E	t	u	N	j	u	k
		t	u	N	j	u	k

Langkah VII: *Pattern* cocok dengan *string*. Karena ada kecocokan, maka algoritma *Knuth-Morris-Pratt* menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*.

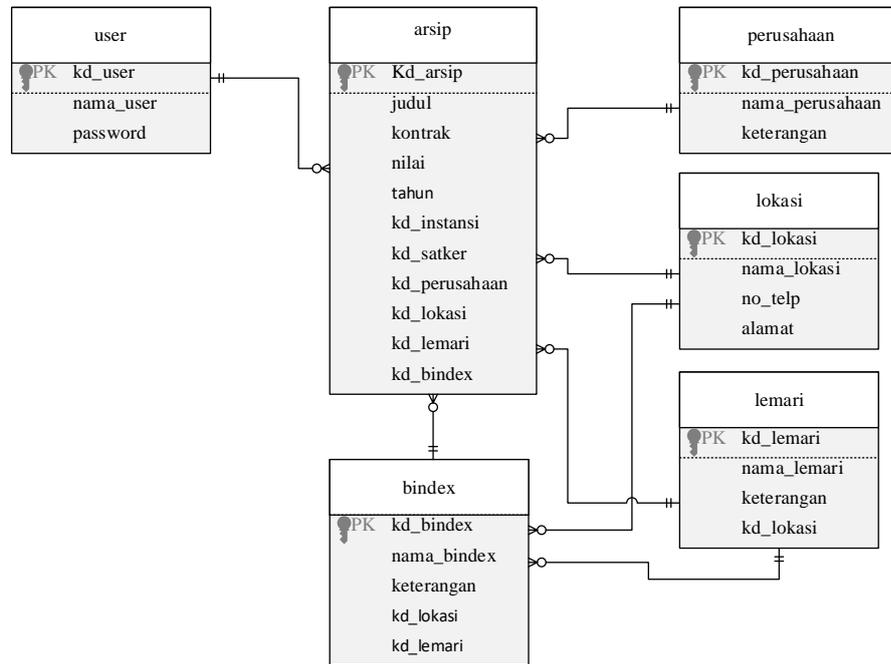
**Tabel 8.** Pencocokan Pattern Dengan String Pada Index Kedelapan

P	e	T	U	N	j	u	k
		T	U	N	j	u	k

Langkah VIII: *Pattern* cocok dengan *string*. Karena ada kecocokan, maka algoritma *Knuth-Morris-Pratt* menyimpan informasi ini, dan *pattern* tidak akan melakukan pergeseran dan melanjutkan pencocokan *pattern* dengan *string*. Namun karena jumlah *pattern* hanya 7 huruf maka pencarian dihentikan dan dapat diperoleh hasil bahwa *pattern* P terdapat kecocokan dengan string S sebesar 100%.



**c. Logical Record Structure (LRS)**



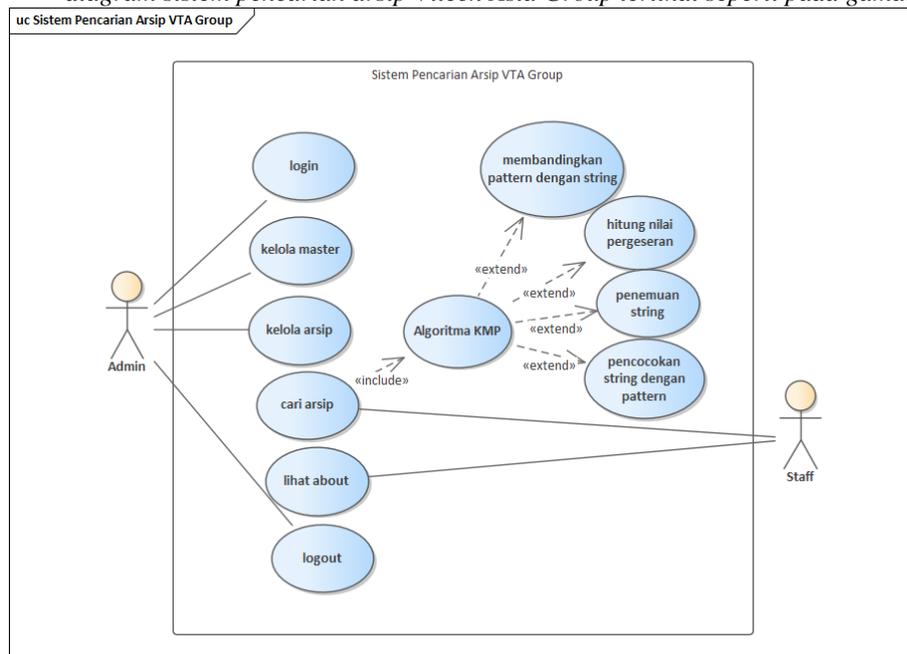
**Gambar 3.** LRS Sistem Pencarian Arsip VTA Group

**3.2.2 Perancangan Unified Modeling Language (UML)**

Perancangan *unfied modelng language (uml)* meliputi perancangan *Use Case Diagram*, *Activity Diagram*, *Squence Diagram* dan *Class Diagram*.

**a. Use Case Diagram**

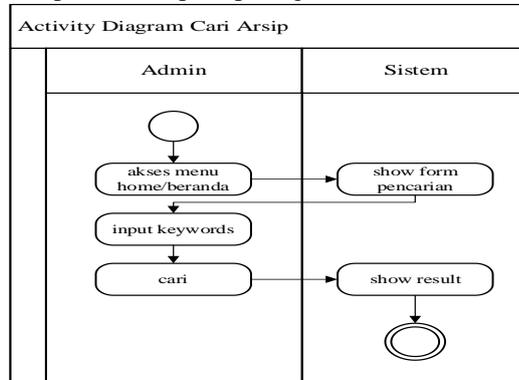
*Use case diagram* menggambarkan hubungan antara aktor dengan sistem. *Use case diagram* sistem pencarian arsip Vitech Asia Group terlihat seperti pada gamabr 4.



**Gambar 4.** Use Case Diagram Sistem Pencarian Arsip

**b. Activity Diagram**

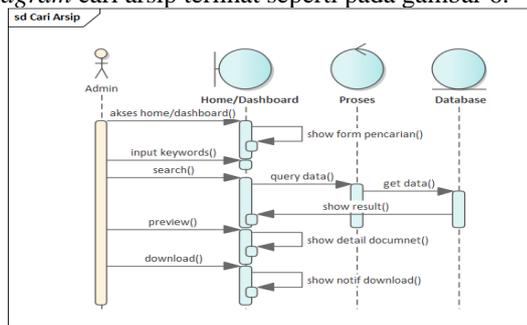
Pada *activity diagram* ini menjelaskan tentang pencarian suatu arsip dokumen. *Activity diagram* cari arsip terlihat seperti pada gambar 5.



**Gambar 5. Activity Diagram Sistem Pencarian Arsip**

**c. Sequence Diagram**

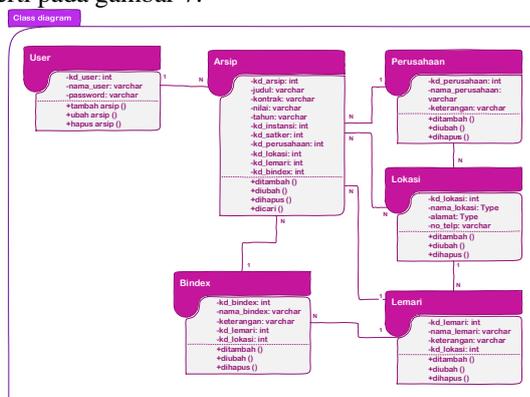
Pada *sequence diagram* ini menjelaskan tentang pencarian suatu arsip dokumen. *sequence diagram* cari arsip terlihat seperti pada gambar 6.



**Gambar 6. Sequence Diagram Sistem Pencarian Arsip**

**d. Class Diagram**

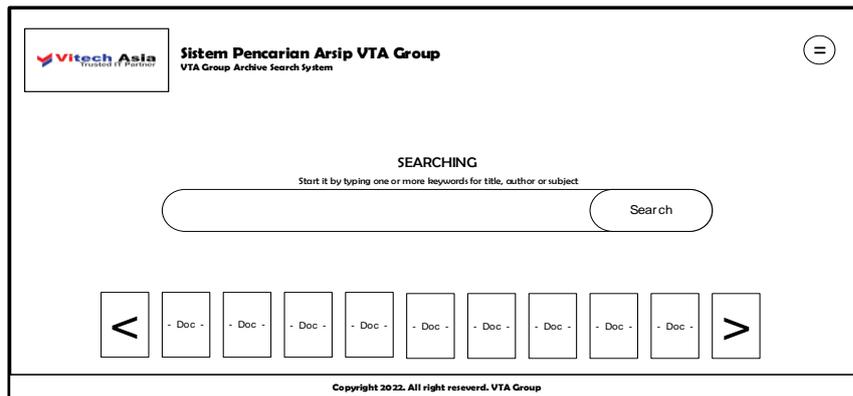
*Class diagram* atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Sukamoto & Shalahuddin, 2018:141). *Class diagram* juga dapat memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain (logical view) dari suatu sistem (T & Syarifuddin, 2020:199). Berikut adalah *class diagram* pada sistem pencarian arsip VTA Group terlihat seperti pada gambar 7.



**Gambar 7. Class Diagram Sistem Pencarian Arsip**

### 3.2.3 Perancangan *User Interface*

Sebagai yang telah dijelaskan pada perancangan *use case* di atas, pada aplikasi sistem pencarian arsip dokumen ini terdapat dua aktor yaitu admin dan staff. Pada rancangan *user interface* cari arsip ini merupakan halaman yang pertama muncul setelah user mengakses sistem. Pada halaman ini juga terdapat *form* pencarian suatu arsip dokumen, di mana jika staff akan melakukan pencarian arsip hanya perlu memasukkan kata kunci lalu klik *button search*. Halaman rancangan *user interface* cari arsip terlihat seperti pada gambar 8.



Gambar 8. Perancangan Sistem Pencarian Arsip VTA Group

## 4. IMPLEMENTASI

Berikut adalah hasil penerapan algoritma *knuth-morris-pratt* dan model *waterfall* pada perancangan aplikasi sistem pencarian arsip vta group yang sebelumnya telah dijelaskan pada bab 2 dan juga telah dilakukan uji serta diperiksa secara menyeluruh untuk mengetahui apakah masih terdapat *bug* atau kesalahan pada sistem. Berikut tampilan halaman pencarian yang ada pada *dashboard*.

### 1. Tampilan Halaman *Dashboard/Home*

Halaman *dashboard/home* merupakan halaman yang pertama muncul setelah admin mengakses sistem. Pada halaman ini juga terdapat *form* pencarian suatu arsip dokumen, di mana jika admin akan melakukan pencarian arsip tinggal memasukkan kata kunci lalu klik *button search*. Halaman *dashboard/home* terlihat seperti pada gambar 9.



Gambar 9. Tampilan Halaman Pencarian

## 5. KESIMPULAN

Berdasarkan analisa, desain/perancangan, hasil dan penerapan metode untuk menyelesaikan masalah pada penelitian ini maka dapat diambil kesimpulan bahwa Dengan adanya sistem pencarian arsip dokumen ini maka aktivitas pencarian dokumen menjadi lebih efektif, cepat, tepat dan efisien, hal ini dibuktikan dengan hasil wawancara *user*, yang sebelumnya memerlukan waktu lebih dari 5 menit untuk mencari suatu dokumen arsip menjadi  $\pm 1$  menit dengan menggunakan sistem pencarian yang telah penulis bangun dan kemungkinan kehilangan atau kerusakan suatu arsip dokumen dapat diminimalisir karena sudah terdapat salinan secara elektronik yang terdapat pada sistem.

## REFERENCES

- Firdaus, N., & Irfan, D. (2020). Rancang Bangun Sistem Informasi Arsip Berbasis Web Menggunakan Framework Codeigniter. *Jurnal Vocational Teknik Elektronika dan Informatika (JoteTEKNIKA)*, VIII(1), 44-52.
- Hidayat, R. (2017). Aplikasi Penjualan Jam Tangan Secara Online Studi Kasus: Toko JAMBORESHOP. *Jurnal Teknik Komputer*, III(2), 90-96.
- Matondang, Z. A. (2018). Implementasi Algoritma String Matching Pencarian Kata Dari Makna Rambu Lalu Lintas Berbasis Android. *Jurnal Sistem Informasi Kaputama (JSIK)*, II(1), 101-106.
- Mualim, W., & Putra, G. U. (2017). Implementasi Framework MVC Pada Sistem Informasi Akademik Di STMIK Yadika Bangil. *Jurnal SPIRIT*, IX(2), 35-39.
- Nurmalasari, Anna, & Arissusandi, R. (2019). Rancang Bangun Sistem Informasi Akuntansi Laporan Laba Rugi Berbasis Web Pada PT United Tractors Pontianak. *Evolusi: Jurnal Sains dan Manajemen*, VII(2), 6-14.
- Nursobah, & Pahrudin, P. (2019). Penerapan Algoritma Pencarian Knuth-Morris-Pratt (KMP) Dalam Sistem Informasi Perpustakaan SMK TI Pratama. *SEBATIK*, XXIII(1), 112-115.
- Ramadhani, D. (2017). Perancangan aplikasi pencarian Buku Pada Perpustakaan Islamic Internasional School Darul Ilmi Murni Dengan Algoritma Knuth Morris Pratt. *Pelita Informatika Budi Darma*, XVI(7), 25-27.
- Rofiq, N., Perdananto, A., & Nurjaya. (2021). Penerapan Model Waterfall Pada Aplikasi Bank Sampah. *INFOTECH: Journal of Technology Information*, VII(1), 19-26.
- Sa'diah, H. T. (2107). Implementasi Algoritma Knuth-Morris-Pratt Pada Fungsi Pencarian Judul Tugas Akhir Repository. *Jurnal Komputasi*, XIV(1), 115-124.
- Sari, D. P., & Wijanarko, R. (2019). Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus Di Rumah Kamera Semarang). *INFORMATIKA DAN RPL*, II(1), 32-36.
- Simanjuntak, H., Lumbantoruan, R., Banjarnahor, W., Sitorus, E., Panjaitan, M., & Panjaitan, S. (2017). Penilaian Kesamaan Entity Relationship Diagram dengan Algoritme Tree Edit Distance. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, VI(1), 15-24.
- Sugiyono. (2017). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D – MPKK*. Bandung: Alfabeta.
- Sulianta, F. (2107). *Teknik Perancangan Arsitektur Sistem Informasi*. Yogyakarta: Andi.
- T, B. K., & Syarifuddin. (2020). Perancangan Sistem Aplikasi Pemesanan Makanan Dan Minuman Pada Cafeteria No Caffe Di Tanjung Balai Karimun Menggunakan Bahasa Pemrograman PHP Dan MySQL. *Jurnal TIKAR*, I(2), 192-206.