

IMPLEMENTASI METODE STRING MATCHING UNTUK APLIKASI PENGARSIPAN DOKUMEN (STUDI KASUS : SMPN 3 SUMBER KAB. CIREBON)

Ida Mulyawati¹, Ridho Taufiq Subagio², Deny Martha³
Sekolah Tinggi Manajemen Informatika dan Komputer CIC Cirebon
Jl.Kesambi 202, Kota Cirebon, Jawa Barat. Telp : (0231) 220250
Email : lena.magdalena@cic.ac.id, abdul3169@yahoo.co.id

Abstrak

Sejalan dengan perkembangan teknologi informasi dan memasyarakatnya internet di masyarakat, maka kebutuhan akan pendidikan bermutu dengan dukungan pemanfaatan teknologi informasi semakin tinggi. Kelebihan sistem pendidikan dengan menggunakan sistem ini akan memicu terjadinya pemerataan dan peningkatan kualitas pendidikan khususnya untuk level pendidikan tinggi dimana kebutuhan akan komputer dan internet sudah merupakan kebutuhan dasar dalam proses pembelajaran. Paper ini membahas tentang standar kebutuhan suatu aplikasi e-Learning dengan memanfaatkan open source dalam rangka menekan biaya serendah mungkin tanpa menurunkan kualitas system secara keseluruhan.

Kata kunci: e-Learning, internet, teknologi informasi, open source, collaboration

Abstract

The progress of information technology and the internet popularization in public, so the need for increasing the education quality with the support of information technology to be higher. Advantages of the educational system by using this system will lead to equity and improving the quality of education, especially for the higher education level such as university where demand for computer and internet is already a basic requirement in the learning process. This paper discussed the need for a standard e-Learning applications by leveraging open source in order to keep costs as low as possible without lowering the quality of the system as a whole.

Keywords: e-Learning, internet, information technology, open source, collaboration

1. PENDAHULUAN

1.1. Latar Belakang Masalah

Sistem pengarsipan pada Sekolah Menengah Pertama Negeri (SMPN) 3 Sumber saat ini masih dilakukan dengan menyimpan arsip seperti data kepegawaian, dan data surat masih berbentuk fisik berupa dokumen dan penyimpanan masih menggunakan lemari arsip. Hal ini menjadikan ruang penyimpanan arsip memakan tempat yang cukup besar, pencarian data membutuhkan waktu yang relatif lama sehingga penyajian informasi menjadi kurang efektif, selain itu semakin banyak arsip yang ditumpuk dapat menyebabkan cepat mengalami kerusakan.

Metode *string matching* yang merupakan bagian dalam proses pencarian *string* memegang peranan penting untuk mendapatkan dokumen yang sesuai dengan kebutuhan informasi dengan lebih cepat. Terdapat algoritma yang digunakan dalam metode string matching salah satunya adalah algoritma Knuth-Morris-Pratt.

Untuk mengatasi permasalahan yang ada pada SMP Negeri 3 Sumber mengenai pengelolaan data arsip dengan menggunakan metode *string matching*, penulis tertarik membuat skripsi dengan judul "*Implementasi Metode String Matching Untuk Aplikasi Pengarsipan Dokumen (Studi Kasus : SMPN 3 Sumber Kab. Cirebon)*".

1.2. Identifikasi Masalah

1. Staff TU mengalami kesulitan dalam melakukan pencarian arsip, dikarenakan harus memeriksa satu per satu arsip yang tersimpan.

2. Bagaimana mengimplementasikan metode *string matching* menggunakan algoritma Knuth-Morris-Pratt untuk aplikasi pengarsipan?
3. Membutuhkan ruang penyimpanan yang luas untuk data arsip.

1.3. Batasan Masalah

1. Aplikasi ini menggunakan algoritma Knuth Morris Pratt.
2. Aplikasi ini digunakan oleh Staff Tata Usaha (TU) yang bertugas untuk mengelola arsip dengan fungsionalitas dalam aplikasi sebagai berikut:
 - a. Kelola Arsip, fitur yang berfungsi untuk mengelola data arsip seperti data pegawai dan data surat.
 - b. Pencarian, merupakan fitur dimana *user* dapat melakukan pencarian dengan algoritma di dalamnya sehingga dapat memudahkan *user*.
 - c. Info, merupakan fitur dimana *user* dapat melihat informasi tentang aplikasi dan logout.
3. Aplikasi ini dikelola menggunakan desktop dengan bahasa pemrograman java dan *database* MySQL.
4. Pencarian arsip berdasarkan jenis arsip, tahun arsip, dan kata yang dicari, untuk data pegawai berdasarkan nama dan untuk data surat berdasarkan perihal.

1.4. Tujuan

1. Membangun aplikasi pengarsipan berbasis desktop yang dapat membantu staff TU dalam pengarsipan.
2. Mempermudah staff TU dalam melakukan pencarian arsip.
3. Penghematan untuk ruang penyimpanan arsip.

2. LANDASAN TEORI

2.1. String Matching

2.1.1. Definisi String Matching

Pencocokan *string* (*string matching*) menurut *Dictionary of Algorithms and Data Structures, National Institute of Standards and Technology* (NIST), diartikan sebagai sebuah permasalahan untuk menemukan pola susunan karakter *string* di dalam *string* lain atau bagian dari isi teks.

2.1.2. Exact Matching

Exact matching merupakan bagian dari *string matching* yang digunakan untuk menemukan *pattern* yang berasal dari satu teks. Contoh pencarian *exact matching* adalah pencarian “pelajar” dalam kalimat “Saya seorang pelajar” atau “Saya seorang siswa”. Sistem akan memberikan hasil bahwa kalimat pertama mengandung kata “pelajar” sedangkan kalimat kedua tidak, meskipun kenyataannya pelajar dan siswa adalah kata yang bersinonim.

2.1.3. Kerangka String Matching

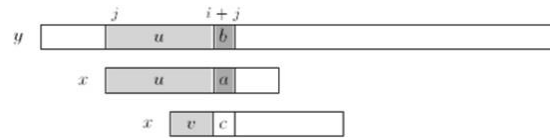
Persoalan pencarian *string* dirumuskan sebagai berikut:

1. Sebuah teks, yaitu *string* dengan panjang n karakter.
2. Pattern, yaitu *string* dengan panjang m .

Dengan sebuah nilai karakter ($m < n$) yang dicari dalam teks. Dalam *string matching*, teks diasumsikan berada di dalam memori, sehingga bila dilakukan pencarian *string* di dalam sebuah arsip, maka semua isi arsip akan dibaca terlebih dahulu kemudian disimpan dalam memori. Jika *pattern* muncul lebih dari sekali dalam teks, maka pencarian hanya akan memberikan keluaran berupa lokasi ditemukan pertama kali.

2.1.4. Algoritma Knuth Morris Pratt

Algoritma Knuth-Morris-Pratt merupakan salah satu algoritma pencarian *string*, yang dikembangkan secara terpisah oleh Donald E. Knuth pada tahun 1967 dan James H. Morris bersama Vaughan R. Pratt pada tahun 1966, kemudian dipublikasikan secara bersamaan pada tahun 1977. Algoritma Knuth-Morris-Pratt melakukan perbandingan karakter teks dan karakter pada pola dari kiri ke kanan. Ide dari algoritma ini adalah bagaimana memanfaatkan karakter-karakter pola yang sudah diketahui ada di dalam teks sampai terjadinya ketidakcocokkan untuk melakukan pergeseran.



Gambar 1. Sumber: Christian Charas & Thierry Lecroq, Handbook of Exact String-Matching Algorithms.

2.1.5. Cara Kerja Algoritma Knuth-Morris-Pratt

Secara sistematis, langkah-langkah yang dilakukan algoritma Knuth-Morris-Pratt pada saat mencocokkan string :

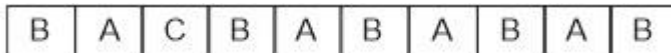
1. Algoritma Knuth-Morris-Pratt mulai mencocokkan *pattern* pada awal teks.
2. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter *pattern* dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi:
 - a. Karakter di *pattern* dan di teks yang dibandingkan tidak cocok (*mismatch*).
 - b. Semua karakter di *pattern* cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.
3. Algoritma kemudian menggeser *pattern* berdasarkan tabel, lalu mengulangi langkah 2 sampai *pattern* berada di ujung teks.

Algoritma ini menemukan semua kemunculan dari *pattern* dengan panjang n di dalam teks dengan panjang m dengan kompleksitas waktu $O(m+n)$. Algoritma ini hanya membutuhkan $O(n)$ ruang dari *memory internal* jika teks dibaca dari file eksternal. Semua besaran O tersebut tidak tergantung pada besarnya ruang *alphabet*.

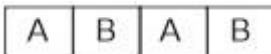
Berikut contoh pencocokan *pattern* dengan menggunakan algoritma Knuth-Morris-Pratt.

Contoh:

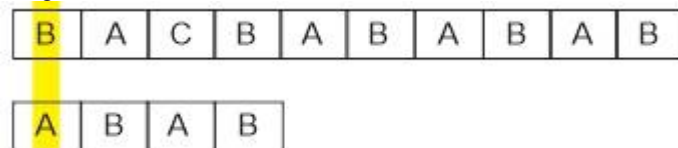
Teks :



Pattern :

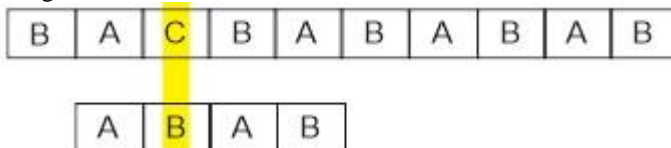


Langkah 1:



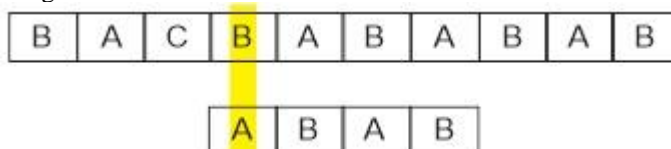
Keterangan: *Pattern* 1 tidak cocok dengan teks 1 maka akan dilakukan pergeseran.

Langkah 2:



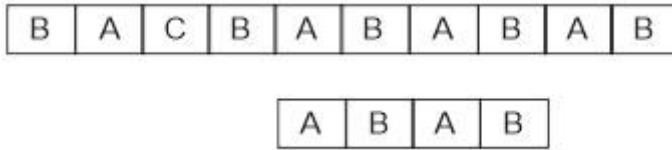
Keterangan: *Pattern* 2 tidak cocok dengan teks 3 maka akan diberlakukan pergeseran.

Langkah 3:



Keterangan: *Pattern* 1 tidak cocok dengan teks 4 maka akan diberlakukan pergeseran.

Langkah 4:



Keterangan: *Pattern* 1 sampai 4 cocok dengan teks 5 sampai 9 maka tidak di perlukan lagi pergeseran. Berdasarkan contoh diatas maka dapat disimpulkan bahwa informasi yang digunakan untuk melakukan pergeseran adalah berdasarkan hitungan ketidakcocokan *Pattern* dari kiri pada teks. (Buulolo,Efori. 2013. *Implementasi Algoritma String Matching Dalam Pencarian Surat dan Ayat Dalam Bible Berbasis Android*. Jurnal Pelita Informatika Budi Darma Vol. 3.)

3. ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

4.1.

3.1.1 Analisis objek penelitian

Sekolah Menengah Pertama Negeri (SMPN) 3 Sumber kabupaten Cirebon adalah salah satu sekolah negeri di kabupaten Cirebon yang sebelumnya merupakan cabang dari SMPN Cirebon Selatan. Pada tahun 2003 barulah Sekolah Menengah Pertama Negeri (SMPN) 3 Sumber diresmikan oleh dinas pendidikan kabupaten Cirebon.

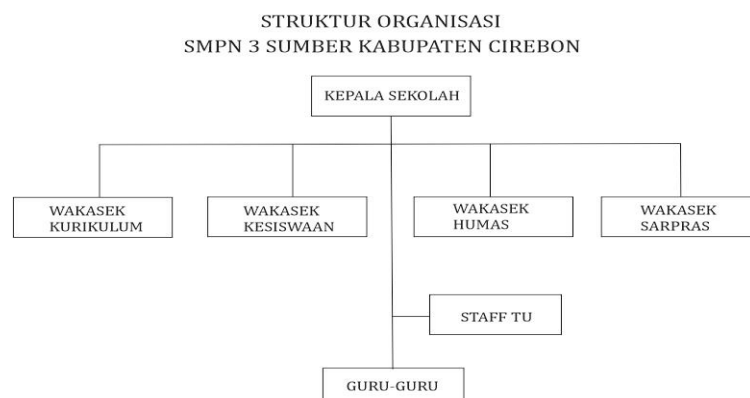
3.1.2. Analisis Masalah

Proses observasi yang dilakukan oleh penulis dengan tujuan mengetahui masalah yang dihadapi sekolah dalam mengelola arsip. Berdasarkan hasil observasi diperoleh informasi bahwa terdapat permasalahan yang dihadapi sekolah dalam mengelola arsip, sebagai berikut:

1. Pencarian data membutuhkan waktu yang relatif lama sehingga penyajian informasi menjadi kurang efektif.
2. Fasilitas penyimpanan arsip tidak sebanding dengan jumlah arsip dan dokumen yang banyak dan selalu bertambah.

Berdasarkan permasalahan yang dihadapi sekolah, maka dibutuhkan solusi yang diharapkan dapat mengatasi masalah dalam mengelola dan pencarian arsip.

Saat ini Sekolah Menengah Pertama Negeri (SMPN) 3 Sumber dipimpin oleh kepala sekolah Dadang Raiman, S.Pd.,MM. dengan jumlah guru sebanyak 52 orang dan staff TU 7 orang dengan susunan organisasi sebagai berikut:

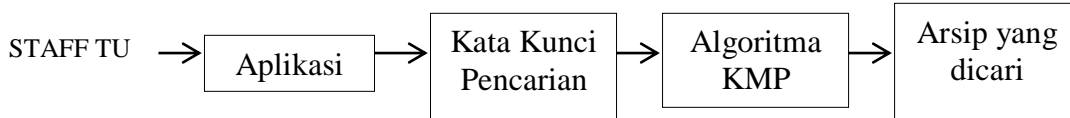


Gambar 2. Struktur Organisasi

3.1.3. Analisis Algoritma Knuth-Morris-Pratt (KMP)

Pada algoritma Knuth-Morris-Pratt proses yang dilakukan adalah dengan memelihara informasi yang digunakan untuk melakukan jumlah pergeseran karena itulah algoritma ini membutuhkan pinggiran. Pinggiran berfungsi untuk memutuskan seberapa jauh *pattern* harus melakukan pergeseran ketika terjadi ketidakcocokan. Rumus untuk melakukan pergeseran yaitu : $(i - b[i])$. *Border function* KMP dapat dilakukan sebelum algoritma KMP dijalankan (*pre-processing*). *Border function* membutuhkan *array of integer* seukuran dengan ukuran *pattern* (sebesar *m*). Algoritma untuk melakukan *pre-process border function* berjalan dalam kompleksitas $O(m)$.

Adapun Alur Algoritma KMP adalah sebagai berikut:



Gambar 3. Alur Algoritma KMP

Pada algoritma Knuth-Morris-Pratt proses yang dilakukan adalah dengan memelihara informasi yang digunakan untuk melakukan jumlah pergeseran karena itulah algoritma ini membutuhkan pinggiran. Pinggiran berfungsi untuk memutuskan seberapa jauh *pattern* harus melakukan pergeseran ketika terjadi ketidakcocokan. Rumus untuk melakukan pergeseran yaitu : $(i - b[i])$

Berikut adalah contoh tabel pinggiran untuk *pattern* RAI untuk data pegawai

I (Indeks)	0	1	2
Pattern	R	A	I
b[i]	-1	0	0

Berikut ini contoh kasus penerapan algoritma Knuth-Morris-Pratt (KMP).

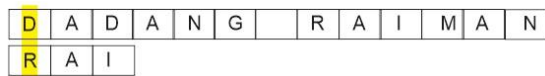
Border function atau pinggiran

Pattern : RAI

b[i] : -1 0 0

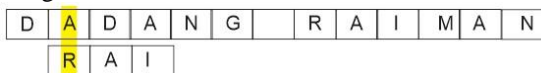
Pencocokan *string*:

Langkah 1



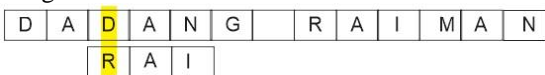
Keterangan: *pattern* 0 dan teks 1 tidak cocok, maka dilakukan pergeseran sebanyak 1 langkah $(i - b[i] = 0 - -1 = 1)$.

Langkah 2



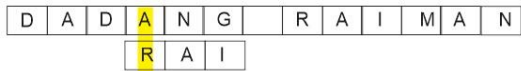
Keterangan: *pattern* 0 dan teks 2 tidak cocok, maka dilakukan pergeseran sebanyak 1 langkah $(i - b[i] = 0 - -1 = 1)$.

Langkah 3



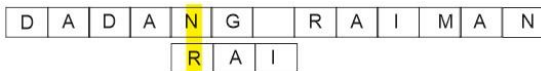
Keterangan: *pattern* 0 dan teks 3 tidak cocok, maka dilakukan pergeseran sebanyak 1 langkah ($i - -b[i] = 0 - -1 = 1$).

Langkah 4



Keterangan: *pattern* 0 dan teks 4 tidak cocok, maka dilakukan pergeseran sebanyak 1 langkah ($i - -b[i] = 0 - -1 = 1$).

Langkah 5



Keterangan: *pattern* 0 dan teks 5 tidak cocok, maka dilakukan pergeseran sebanyak 1 langkah ($i - -b[i] = 0 - -1 = 1$).

Langkah 6



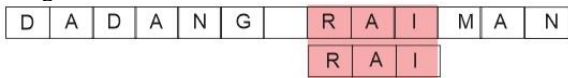
Keterangan: *pattern* 0 dan teks 6 tidak cocok, maka dilakukan pergeseran sebanyak 1 langkah ($i - -b[i] = 0 - -1 = 1$).

Langkah 7



Keterangan: *pattern* 0 dan teks 7 tidak cocok, maka dilakukan pergeseran sebanyak 1 langkah ($i - -b[i] = 0 - -1 = 1$).

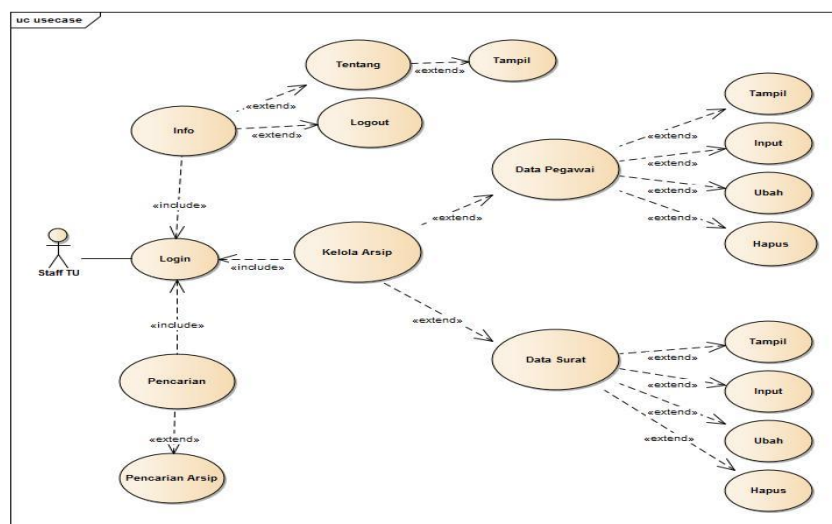
Langkah 8



Keterangan: Teks dan *pattern* cocok.

3.1.4. Analisis Sistem Aplikasi

3.1.4.1. Use Case Diagram



Gambar 4. Usecase Diagram

3.2. Perancangan Sistem

1. Rancangan Form Menu Utama

KELOLA ARSIP PENCARIAN INFO

SEKOLAH MENENGAH PERTAMA NEGERI

LOGO

3 SUMBER KABUPATEN CIREBON

Gambar 4. Form Menu Utama

2. Rancangan Form Pencarian

PENCARIAN

Jenis Arsip

Tahun Arsip

Masukkan Kata

Gambar 5. Form Pencarian

3. Rancangan Form Hasil Pencarian Data Pegawai

Jenis Arsip

Tahun Arsip

Masukkan Kata

Menu Utama

NIP	NAMA	TMT	TUGAS TAMBAHAN	PANGKAT	ALAMAT	NO. HP
19741120 200083 1 000	Dadang Raiman, S.Pd.,MM	6 Maret 2015	Kepala Sekolah	IV / A	Blok Tengah Rt. 04 / Rw. 02 Ds. Gempol Kec. Gempol Kab. Cirebon	082317086303

Gambar 6. Form Hasil Pencarian Data Pegawai

4. Rancangan Form Hasil Pencarian Data Surat

The screenshot shows a search results form with the following fields and controls:

- Jenis Arsip: Data Surat (dropdown menu)
- Tahun Arsip: 2016 (text input)
- Masukkan Kata: Tug (text input)
- Tampilkan (button)
- Menu Utama (button)

No. Surat	Tanggal Surat	Kode Klasifikasi	Perihal	Jenis Surat	Keterangan	Tanggal Arsip
800/347/SMP03	8 Mei 2016	800	Surat Tugas	Surat Keluar	Pengawas UN	31 Mei 2016

Gambar 7. Form Hasil Pencarian Data Pegawai

4. IMPLEMENTASI DAN PENGUJIAN SISTEM

1. Form Menu Utama

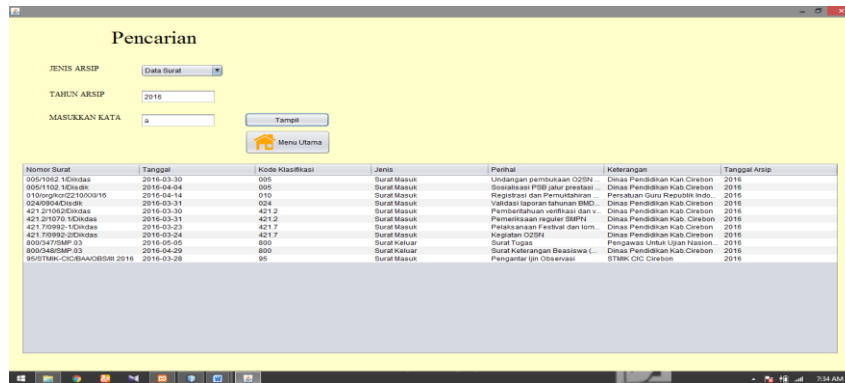


Gambar 8. Form Menu Utama

Keterangan:

Form Menu Utama berisi pilihan-pilihan menu yang dapat digunakan oleh staff TU. Di dalam form menu utama terdapat menu kelola arsip yang berfungsi untuk mengelola data pegawai dan data surat, pencarian berfungsi dalam pencarian arsip, dan info berisi tentang aplikasi dan logout.

2. Form Pencarian Arsip

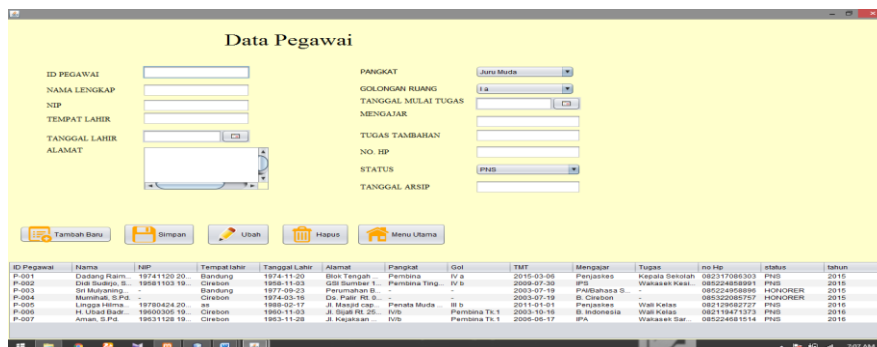


Gambar 9. Form Pencarian Arsip

Keterangan:

Form pencarian arsip adalah bagian dari menu pencarian. Staff TU dapat melakukan pencarian arsip dengan memilih jenis arsip, tahun arsip dan memasukkan kata yang dicari, dengan menggunakan metode *string matching*.

3. Form Data Pegawai

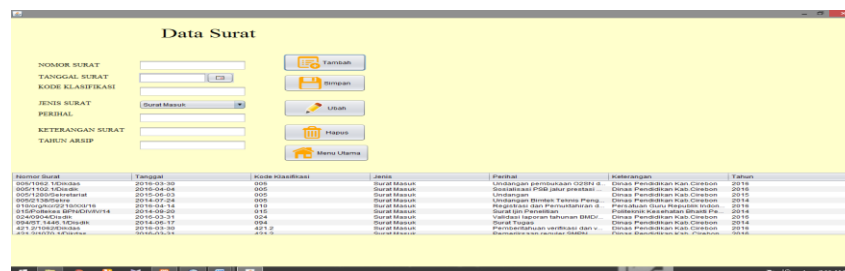


Gambar 10. Form Data Pegawai

Keterangan:

Form data pegawai adalah bagian dari menu kelola arsip. Staff TU dapat melakukan tambah baru, simpan, ubah dan hapus data pegawai.

4. Form Data Surat



Gambar 11. Form Data Surat

Keterangan:

Form data surat adalah bagian dari menu kelola arsip. Staff TU dapat melakukan tambah baru, simpan, ubah dan hapus data pegawai.

4.3.2 Pengujian Black Box

1. Form Login

Tabel 1. Pengujian form login

No	Jenis	Pengujian	Hasil yang diharapkan	Output	Hasil Uji
1.	Login	Input data username = "admin" password = "12345"	Tampil halaman menu utama	Tampil halaman menu utama	Valid
2.	Login	Input data username = "admin" password = "1234"	Tampil pesan kesalahan.	Tampil pesan kesalahan 'password yang anda masukkan salah.'	Valid

Tabel 2. Pengujian form login (Lanjutan)

No	Jenis	Pengujian	Hasil yang diharapkan	Output	Hasil Uji
1.	Login	Input data username= "adm" password= "12345"	Tampil pesan kesalahan	Tampil pesan kesalahan 'username anda salah'	Valid

2. Form Menu Utama

Tabel 3. Pengujian form menu utama

No	Jenis	Pengujian	Hasil yang diharapkan	Output	Hasil Uji
1.	Kelola Arsip	Memilih kelola arsip untuk data	Tampil halaman data pegawai	Tampil halaman data pegawai	Valid

		pegawai.			
2.	Kelola Arsip	Memilih kelola arsip untuk data surat.	Tampil halaman data surat	Tampil halaman data surat	Valid
3.	Pencarian	Memilih pencarian.	Tampil halaman pencarian	Tampil halaman pencarian	Valid
4.	Info	Memilih info untuk tentang	Tampil halaman tentang	Tampil halaman tentang	Valid
5.	Info	Memilih info untuk logout	Tampil halaman logout	Tampil halamanan logout	Valid

3. Form Data Pegawai

Tabel 4. Pengujian form data pegawai

No	Jenis	Pengujian	Hasil yang diharapkan	Output	Hasil Uji
1.	Data Pegawai	ketika memilih data pegawai	Menampilkan form input data pegawai	Tampil form input data pegawai	Valid
2.	Tambah Baru	Ketika memilih tambah data baru.	Dapat menginputkan data baru.	Menginput data baru	Valid
3.	Tambah Baru	masih terdapat data pada <i>textbox</i> .	Hapus data yang terdapat pada <i>textbox</i> .	Hapus data yang terdapat pada <i>textbox</i> .	Valid
4.	Simpan	Input semua data pada <i>textbox</i> .	Dapat Menyimpan dan tampil pesan berhasil	Tampil pesan 'Berhasil di simpan'	Valid
5.	Simpan	Input data pegawai pada <i>textbox</i> dan diisi dengan data yang sudah ada.	Tampil pesan gagal simpan	Tampil pesan 'Gagal dalam penyimpanan'	Valid
6.	Ubah	Mengubah data pegawai	Tampil pesan data berhasil diubah	Tampil pesan 'data berhasil diubah'	Valid
7.	Hapus	Menghapus data pegawai	Dapat Mengapus data pegawai.	Data akan terhapus	Valid

5. KESIMPULAN

Setelah dilakukan analisa program, maka dapat diperoleh kesimpulan sebagai berikut :

1. Aplikasi dapat mencari arsip berdasarkan jenis arsip, tahun dan kata yang di cari.
2. Algoritma Knuth-Morris-Prat telah berhasil di implementasikan pada aplikasi pengarsipan.
3. Berdasarkan uji coba aplikasi, algoritma KMP dapat membantu user untuk melakukan pencarian arsip dengan mudah.

DAFTAR PUSTAKA

- [1] Buulolo,Efori. 2013. *Implementasi Algoritma String Matching Dalam Pencarian Surat dan Ayat Dalam Bible Berbasis Android*. Jurnal Pelita Informatika Budi Darma Vol. 3.
- [2] Charras, C dan Lecroq Thierry. 2004. *Handbook of exact string matching algorithms*. College Publication.
- [3] *Dictionary of Algorithms and Data Structures*.National Institute of Standards and Technology. <http://www.nist.gov/dads/>. Diakses tanggal 22 April 2016.
- [4] Pudjo Widodo, Prabowo dan Herlawati. 2011. *Menggunakan UML*. Yogyakarta: Andi Publisher.
- [5] Sarno, Riyanarto, dkk, 2012. *Semantic Search*. Yogyakarta: Andi Publisher.
- [6] Syaroni,M dan Munir,R. 2004. *Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching) dalam Bahasa Inggris*. Institut Teknologi Bandung (ITB).