

## ANALISIS PERBANDINGAN PENGGUNAAN ALGORITMA SEQUENTIAL SEARCH DAN BINARY SEARCH PADA APLIKASI SURAT PERJALANAN DINAS

Dian Markuci, Cahyo Prianto

Program Studi D4 Teknik Informatika  
Politeknik Pos Indonesia, Jalan Sariosih No.54 Bandung, Indonesia  
*dianmarkucii@gmail.com*

### ABSTRAK

Algoritma pencarian merupakan proses menemukan data atau informasi tertentu dalam sekumpulan data menggunakan kata kunci atau *keyword*. Terdapat bermacam-macam algoritma pencarian dan setiap algoritma masing-masing mempunyai kelebihan dan kekurangan. Penelitian kali ini penulis akan meneliti dua algoritma yaitu *sequential search* dan *binary search*. Pada studi kasus ini dua algoritma tersebut akan diterapkan untuk pencarian data pegawai dari keseluruhan data sebanyak 250 data pegawai pada aplikasi surat perjalanan dinas. kemudian akan dianalisis kecepatan rata-rata waktu pencarian masing-masing algoritma. Selanjutnya akan dilakukan perbandingan dua algoritma tersebut dengan melakukan simulasi proses pencarian di beberapa *device*. Setelah melakukan simulasi didapatkan kecepatan rata-rata pencarian *sequential search* untuk *device 1*, *device 2*, dan *device 3* yang di peroleh yaitu 0.00165 detik, 0.00065 detik, 0.00102 detik. Sedangkan kecepatan pencarian *binary search* untuk *device 1*, *device 2*, dan *device 3* yang di peroleh yaitu 0.00012 detik, 0.00011 detik, 0.00017 detik. Berdasarkan pengujian parameter waktu kecepatan pencariannya dapat disimpulkan bahwa algoritma *binary search* memiliki performa kecepatan pencarian yang lebih baik dibandingkan dengan algoritma *sequential search*. Algoritma *binary search* cenderung lebih stabil dan membutuhkan waktu yang sedikit dalam melakukan proses pencarian data. Sehingga *binary search* akan lebih cocok untuk diterapkan pada aplikasi surat perjalanan dinas atau pencarian data pada aplikasi dengan jumlah data yang banyak atau besar.

**Kata kunci :** *algoritma, sequential search, binary search, kecepatan, aplikasi, keyword.*

### 1. PENDAHULUAN

Perkembangan teknologi web kini begitu pesat dibandingkan dengan teknologi lainnya di bidang yang sama, pembuatan aplikasi dan website menjadi hal yang sudah sangat umum. Banyaknya penggunaan internet di dunia, akan mempengaruhi kehidupan sehari-hari. Penggunaan mesin pencari atau *search engine* mempermudah pengguna dalam memperoleh beragam informasi contohnya *Google*. Sistem pencarian menjadi salah satu fitur yang dibutuhkan pada sebuah website ataupun aplikasi agar lebih efektif. sistem pencarian memiliki algoritma di dalamnya dan algoritma *search engine* juga sangat banyak. *Search engine* merupakan suatu sistem perangkat lunak komputer yang dibuat untuk mempermudah penggunaanya dalam mencari dan menemukan informasi. Mesin pencari mempunyai keahlian dalam mencari atau menelusuri berbagai informasi yang dibutuhkan dan sudah dipublikasikan hanya dengan menulis *keyword* atau kata kunci, dengan begitu informasi yang berkaitan dengan *keyword* tersebut akan ditelusuri mesin pencari secara otomatis mengenai sampai pengguna dapat menemukan informasi secara mudah [1]. Algoritma akan menyelesaikan persoalan menggunakan prosedur yang jelas [2]. Algoritma merupakan langkah komputasi yang dapat di implementasikan dalam membuat program [3]. Dari banyaknya algoritma yang ada tentu setiap algoritma memiliki kelebihan dan kekurangan maka penting dalam mencapai tujuan yang diinginkan perlu mengetahui kelebihan dan kekurangan algoritma yang akan digunakan.

Setelah melakukan observasi penulis menemukan adanya indikasi permasalahan di dinas XYZ yaitu salah satunya mengenai pengelolaan surat perjalanan dinas. Pengolahan data surat perjalanan dinas membutuhkan waktu yang lama dalam proses pengelolaannya karena dalam prosesnya masih dilakukan dengan cara manual. Permasalahan yang sering terjadi yaitu penyimpanan berkas dokumen terlalu banyak sehingga pencarian data menjadi sebuah kendala. Berdasarkan permasalahan yang ada diperlukan suatu mekanisme yang terkomputerisasi berupa sebuah sistem yang dapat mempermudah dan meningkatkan kinerja dalam memberikan layanan yang lebih baik dan cepat, baik dalam pembuatan maupun pencarian data pegawai dan laporan surat perjalanan dinas. Penulis akan menerapkan penggunaan algoritma pencarian. Dalam penelitian ini penulis menggunakan dua algoritma yaitu *sequential search* dan *binary search*, kedua algoritma tersebut banyak diimplementasikan pada suatu sistem dan dikenal sederhana penerapannya. Aplikasi surat perjalanan dinas merupakan aplikasi yang datanya akan selalu bertambah, maka dari itu penulis menggunakan dua algoritma ini untuk dijadikan perbandingan, manakah algoritma pencarian yang lebih baik untuk diterapkan pada aplikasi surat perjalanan dinas. Penelitian ini menggunakan data pegawai sebagai data yang dicari pada proses perbandingan.

Perbedaan penelitian yang dilakukan ini dengan penelitian-penelitian sejenis lainnya yaitu terdapat pada proses pengujian. Pada penelitian ini dua

algoritma yaitu *sequential search* dan *binary search* akan diterapkan pada satu halaman untuk pencarian data pegawai di aplikasi surat perjalanan dinas. kemudian akan di analisis berdasarkan parameter kecepatan waktu pencarian masing-masing algoritma. Selanjutnya akan dilakukan perbandingan dua algoritma tersebut dengan melakukan simulasi proses pencarian di beberapa *device* dengan spesifikasi yang berbeda. Selain itu terdapat penambahan grafik perbandingan pada aplikasi yang di bangun untuk menunjukkan perbedaan waktu proses pencarian dari kedua algoritma agar terlihat lebih jelas.

## 2. TINJAUAN PUSTAKA

### 2.1. Algoritma

Dalam buku berjudul “Aljabar wal Muqabala” Algoritma berasal dari kata *al-kawarizmi*. Algoritma dapat di definisikan sebagai susunan langkah penyelesaian permasalahan secara logis dan sistematis. Dalam membangun suatu program dengan kata lain membuat suatu alat bantu untuk menyelesaikan masalah [3]. Algoritma merupakan serangkaian tahapan perhitungan yang mengubah masukan menjadi keluaran [4].

### 2.2. Algoritma Sequential search

Algoritma *Sequential search* atau bisa juga disebut pencarian *linear* atau *linear search* adalah algoritma pencarian sederhana yang pencariannya dilakukan secara beruntun. Algoritma beruntun merupakan proses yang membandingkan tiap elemen larik satu persatu berurutan, mulai dari elemen awal, sampai dengan elemen yang dicari bisa ditemukan [5]. Algoritma ini bekerja dengan mencocokkan data pada *database* dengan data yang dicari pencarian dilakukan secara beruntun dari data awal hingga data terakhir [6].

Langkah-langkah pencarian dengan *sequential search*:

1. Melakukan perbandingan beruntun pada kumpulan data dengan menggunakan keyword atau kata kunci yang akan dicari
2. Pencarian ini melakukan *looping* atau pengulangan dari data pertama hingga jumlah data ke-n
3. Setiap perulangan dilakukan pencocokan data dengan kata kunci yang dicari
4. Apabila terdapat kecocokan data dengan kata kunci yang di cari maka data berhasil ditemukan dan pencarian di hentikan, sebaliknya apabila tidak ada kecocokan data hingga data terakhir maka pencarian dihentikan dan data tidak ditemukan

### 2.3. Algoritma Binary search

*Binary search* adalah algoritma yang dikembangkan dengan ide dasar berbeda untuk menangani masalah pencarian data pada pemrograman [7]. *Binary search* bekerja dengan baik untuk pencarian data yang sudah terurut [8]. Berikut langkah-langkah pencarian *binary search* :

1. Mengurutkan data terlebih dahulu bisa secara *ascending* (dari depan ke belakang) atau *descending* (dari belakang ke depan).
2. Membagi dua *array* atau keseluruhan data. Apabila data yang dicari lebih kecil dari data yang terletak di tengah, maka pencarian data akan berlanjut ke sebelah kiri. Apabila data yang dicari lebih besar dari data terletak ditengah, maka pencarian data akan berlanjut ke sebelah kanan kemudian proses pencarian tersebut akan terus berulang sampai data yang dimaksud atau dicari dapat ditemukan [9]

### 2.4. Penelitian Sebelumnya

Penelitian Yuri Rahmanto, Joni Alfian, Damayanti, Rohmat Indra Borman dengan judul penelitian “Penerapan Algoritma *Sequential search* pada Aplikasi Kamus Bahasa Ilmiah Tumbuhan” pada tahun 2021. Indikasi permasalahan pada penelitian ini yaitu kurangnya ketertarikan siswa mempelajari kamus bahasa yang tebal. Maka dibutuhkan kamus digital untuk mempermudah siswa mempelajari bahasa ilmiah pada tumbuhan. Hasil dari penelitian ini berupa aplikasi android kamus bahasa ilmiah tumbuhan berbasis Android dengan implementasi algoritma *sequential search*. Total data bahasa ilmiah pada *database* sebanyak 480 data. Pengujian kecepatan dilakukan dengan melihat *page load time* menggunakan fungsi *microtime*. membutuhkan waktu rata-rata 0,01 ms [10].

Penelitian Alba Ragil Sutra Deva dengan judul penelitian “Penerapan Algoritma *Binary search* Pada Aplikasi *E-Order*” pada tahun 2021. Indikasi permasalahan pada penelitian ini yaitu proses *order* masih dilakukan manual hal tersebut tentu mempengaruhi kepuasan pelanggan. Maka dari itu dibutuhkan sebuah aplikasi untuk mempermudah pemesanan dan urutan pemesanan pelanggan. Hasil dari penelitian ini berupa aplikasi android *e-order* dengan implementasi algoritma *binary search* untuk mencari menu makanan dan minuman dengan total data menu pada *database* sebanyak 54 data [8].

Penelitian Muhamad Akbar Pratama dengan judul penelitian “Analisis Perbandingan Algoritma *Linear* dan *Binary* pada Aplikasi Kamus Bahasa Indonesia-Jawa” pada tahun 2020. Penelitian ini membahas mengenai penerapan algoritma yaitu *linear*, *binary* dan *query SQL* pada aplikasi kamus bahasa indonesia-jawa. Indikasi permasalahan pada penelitian ini mengenai evaluasi kinerja algoritma *linear* dan algoritma *binary* dalam proses pencarian data berdasarkan *runtime* dan *memory consumption*. *Query sql* memiliki *runtime* 18.169 ms dan *memory consumption* 6.602 mb, *linear* memiliki *runtime* 13.665412 ms dan *memory consumption* 2.027 mb. *binary search* memiliki *runtime* 0.074008 ms dan *memory consumption* 2.073 mb. Sehingga dapat dinilai dalam hal *runtime* algoritma *binary* lebih baik dibanding algoritma *linear* dan *query sql*. Sedangkan

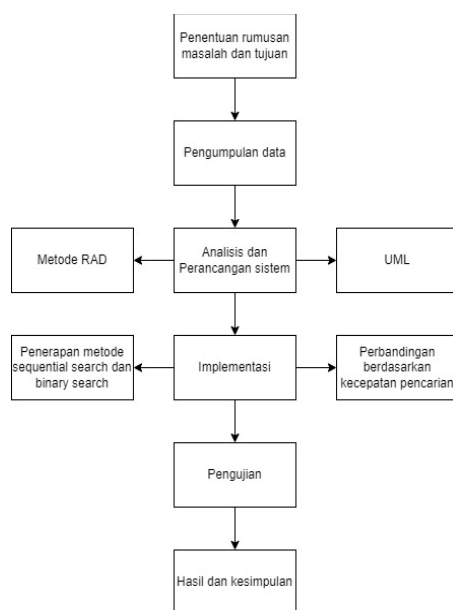
dalam hal *memory consumption* menggunakan lebih banyak dibanding *linear* dan *binary* [11].

Penelitian Nurul Imamah dengan judul Perbandingan Algoritma *Sequential search* Dan Algoritma *Binary search* Pada Aplikasi Bahasa Indonesia Menggunakan PHP Dan JQUERY. Penelitian ini membahas analisis kinerja tiga metode berupa kecepatan dalam proses pencarian data dan memori yang terpakai pada implementasinya. Pengujiannya dilakukan dengan mencari kata di awal, tengah, dan akhi dan dari banyaknya data yang diambil dari unggahan dokumen pdf. Hasil penelitian ini menunjukkan pencarian algoritma *Binary search* lebih cepat dibandingkan *Sequential search* namun mengenai penggunaan memorinya lebih sedikit [5].

Penelitian Yoga Religia dengan judul “Analisis Algoritma *Sequential search* Dan *Binary search* Indikasi permasalahan pada penelitian ini yaitu pencarian pada big data. Algoritma pencarian yang diterapkan adalah *sequential search* dan *binary search*. Penelitian ini menganalisis kinerja dari kedua algoritma tersebut dalam sebuah data besar dengan total 15000 *record*. Setelah melakukan pengujian sebanyak 5 kali didapatkan hasil waktu pencarian, dimana kasus terbaik *sequential search* mendapat rata-rata 0.000942 ms dan kasus terburuk mendapat rata-rata 0.014357 ms. kasus terbaik *binary search* mendapat rata-rata 0.00013 ms dan kasus terburuk mendapat rata-rata 0.00015 ms[12].

### 3. METODE PENELITIAN

Adapun tahap-tahap proses yang akan di lakukan dalam penelitian ini akan digambarkan menggunakan diagram alir berikut ini :



Gambar 1. Diagram Alir Penelitian

Tahapan – tahapan yang dilakukan berdasarkan diagram alir penelitian diatas :

#### 3.1. Menentukan Rumusan Masalah dan Tujuan

Setelah melakukan penelitian di dinas xyz maka dilakukan perumusan masalah sesuai dengan kondisi dan permasalahan yang didapatkan di dinas xyz. Adapun tujuannya akhir pada penelitian ini adalah membangun sebuah aplikasi surat perjalanan dinas dengan penerapan algoritma pencarian yang dinilai lebih baik untuk di terapkan pada aplikasi.

#### 3.2. Pengumpulan Data

Dalam proses perencanaan diperlukan suatu analisis yang sangat teliti. Semakin rumit permasalahan yang dihadapi maka semakin kompleks pula analisisnya. Untuk mendapatkan suatu analisis yang baik dan benar, maka dari diperlukan data/informasi dan teori konsep dasar serta alat bantu yang memadai, sehingga kebutuhan data sangat mutlak diperlukan. Adapun metode perolehan data yang peneliti lakukan dengan cara:

1. Metode literatur yaitu suatu metode yang digunakan untuk mendapatkan data dengan cara mengumpulkan, mengidentifikasi dan mengolah data.
2. Metode observasi yaitu metode yang digunakan untuk mendapatkan data dengan cara melakukan survei langsung ke lokasi. Hal ini sangat diperlukan untuk mengetahui kondisi lokasi yang sebenarnya sehingga dapat menerapkan asumsi-asumsi pendekatan yang sesuai dengan hasil survei dan peninjauan langsung ke lapangan.

#### 3.3. Analisis dan Perancangan Sistem

Analisis dan perancangan sistem merupakan tahap pengembangan sistem pada instansi tempat penelitian. Sistem ini dibuat dengan melihat sistem yang sedang berjalan, dan permasalahan yang ada. Tujuan dari perancangan sistem ini adalah tahap akhir dalam menyelesaikan penelitian yang dilakukan, dengan adanya sistem yang akan dibangun diharapkan pengelolaan surat perjalanan dinas tidak lagi dilakukan secara manual. Metode yang akan digunakan peneliti yaitu metode RAD (*Rapid Application Development*) sebagai metode pengembangan sistem. Berikut adalah penjelasannya:

##### 1. Requirements Planning

Tahap ini mengidentifikasi syarat-syarat informasi dan masalah-masalah yang dihadapi instansi untuk meningkatkan pelayanan.

##### 2. RAD Design Workshop

tahapan ini disebut dengan istilah *workshop*. Tujuan tahapan ini membahas dan menganalisis aplikasi yang dibangun atau dikembangkan, Selama tahapan ini diskusi dengan pihak instansi dilakukan untuk membahas perancangan dan modul-modul yang diusulkan. tahap ini dilakukan untuk mendapatkan *respons* dari pengguna aplikasi yang ada dilingkungan instansi.

3. *Implementation*

Tahapan ini merupakan tahap untuk menganalisis aspek-aspek bisnis dan non teknis instansi. Pada tahap ini menerapkan dan mengembangkan desain program yang sebelumnya sudah dianalisis.

Pada bagian perancangan sistem, akan dibuat rancangan sistem yang detail agar dapat dimengerti oleh user. Rancangan ini terdiri dari 2 tahap :

1. Tahap saat merancang gambaran sistem atau proses yang akan dibangun yang dimodelkan menggunakan diagram UML.
2. Tahap merancang tampilan *user interface* dan struktur menu sistem yang akan digunakan.

3.4. **Implementasi**

Implementasi ini bertujuan menganalisis aspek-aspek bisnis dan non teknis perusahaan serta membahas hasil evaluasi dari uji coba hingga digunakan secara komersil. Selain menerapkan metode RAD, Metode *sequential search* dan *binary search* diterapkan pada aplikasi pada tahap ini bertujuan agar Analisa bisa lebih terukur dalam melakukan proses pencarian data pada aplikasi surat perjalanan dinas.

3.5. **Pengujian**

Setelah melakukan implementasi, langkah yang dikerjakan selanjutnya adalah pengujian terhadap algoritma yang diterapkan pada sistem yang telah dibuat menggunakan fungsi *microtime*. Pengujian bertujuan mengetahui perbandingan kecepatan pencarian algoritma yang diimplementasikan pada sistem.

3.6. **Hasil dan Kesimpulan**

Dari hasil pengujian sistem yang sudah dibangun, maka penulis mampu menentukan kesimpulan yang didapatkan dari proses analisis hingga penerapan metode dalam penelitian yang dilakukan sehingga penulis bisa memberikan saran terhadap sistem yang telah dibuat.

4. **HASIL DAN PEMBAHASAN**

4.1. **Analisis Sistem**

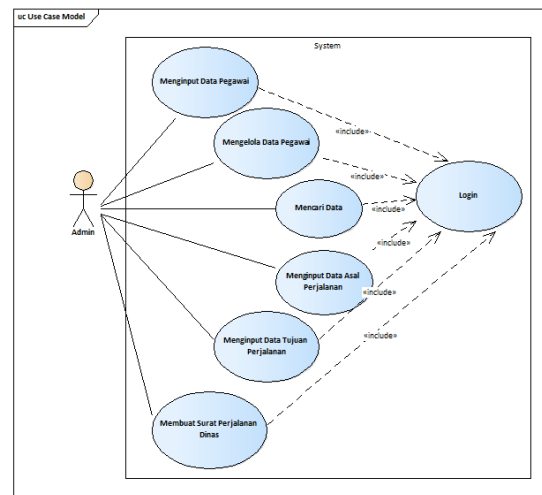
Analisis sistem berjalan terkait pengelolaan surat perjalanan dinas di dinas xyz masih dilakukan secara manual, dimana bagian kepegawaian umum harus menyiapkan blangko surat perjalanan dinas, kemudian menginput satu satu kebutuhan isian data yang terdapat pada blangko. pada pencarian data pegawai untuk keperluan perjalanan dinas membutuhkan waktu yang cukup lama karena proses pencariannya masih dilakukan secara manual dengan mencari satu-persatu data dan menginputnya ke microsoft excel. Kemudian format dokumen yang sudah terisi disimpan dan bisa

untuk di cetak dan dikelola sesuai prosedur yang ada pada dinas xyz.

Analisis sistem yang akan dibangun yaitu sebuah aplikasi berbasis web dengan menerapkan metode *sequential search* dan *binary search* untuk dijadikan perbandingan metode manakah yang lebih baik untuk diterapkan pada aplikasi surat perjalanan dinas. Sistem yang dibangun meliputi login untuk validasi admin sebelum masuk ke dalam sistem. Apabila berhasil, sistem akan menampilkan halaman dashboard dan admin memiliki akses untuk menginput, mengelola, membuat dan mencetak surat perjalanan dinas.

4.2. **Use Case Diagram**

*Use Case* diagram adalah diagram yang menunjukkan suatu kelompok *use case* dan actor serta hubungannya[13]. Berikut ini merupakan *use case* diagram yang menunjukkan interaksi admin dengan sistem yang sudah dirancang.

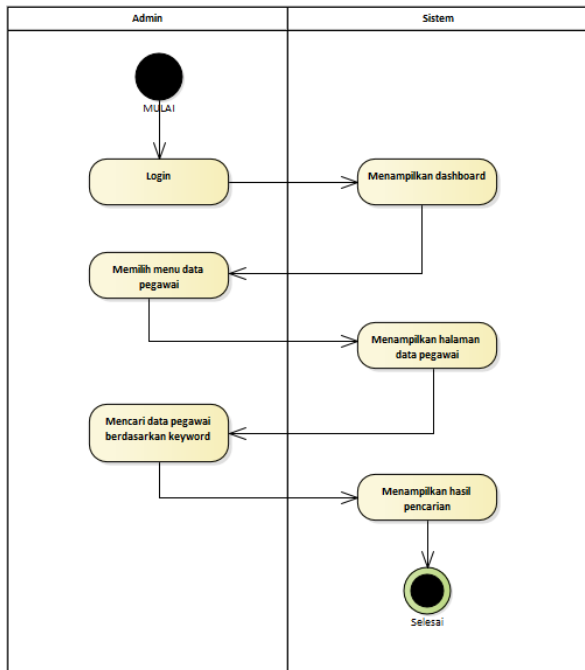


Gambar 2. Use Case Diagram

Pada gambar 2 menunjukkan *use case* diagram yang menjelaskan aktor yaitu admin memiliki akses untuk menginput data pegawai, mengelola data pegawai, mencari data, menginput data asal perjalanan, menginput data tujuan perjalanan, dan membuat surat perjalanan dinas. Namun untuk bisa melakukan semua akses tersebut memerlukan login terlebih dahulu sebagai validasi admin sebelum masuk kedalam sistem.

4.3. **Activity Diagram**

*Activity* diagram adalah diagram yang menjelaskan bentuk alir kerja atau *workflow* atau suatu aktivitas pada sistem [14]. Berikut gambaran aktivitas sistem untuk pencarian data pegawai.



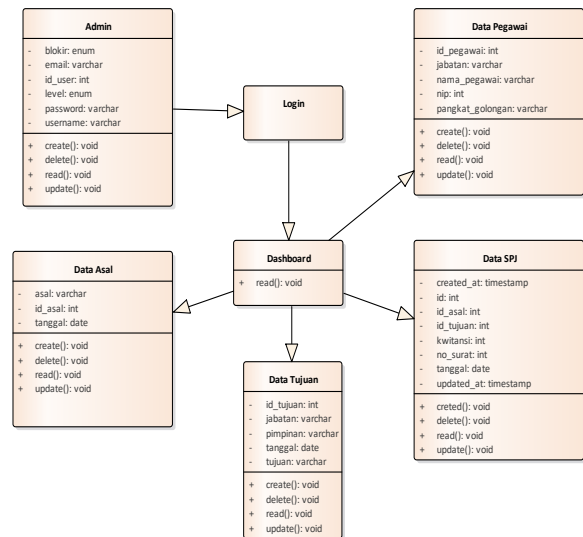
Gambar 3. Activity Diagram

Pada gambar 3 menunjukkan *activity diagram* yang menjelaskan gambaran aktivitas sistem melakukan pencarian data pegawai dimana admin melakukan *login* ke sistem terlebih dahulu, selanjutnya sistem akan menampilkan halaman *dashboard*, kemudian admin memilih menu data pegawai dan sistem akan menampilkan halaman data pegawai. Selanjutnya admin dapat melakukan pencarian data dengan menginputkan *keyword* berupa NIP pegawai setelah itu sistem akan melakukan proses perbandingan antara algoritma *sequential search* dan *binary search* yang sudah diterapkan pada sistem dan hasilnya akan ditampilkan dengan bentuk grafik beserta keterangan waktu pencarian yang dilakukan masing-masing algoritma.

#### 4.4. Class Diagram

Class diagram adalah sebuah diagram yang menjelaskan class yang ada pada sistem dan relasinya secara *logic* [15] Berikut ini merupakan diagram yang menjelaskan class-class yang ada pada aplikasi surat perjalanan dinas beserta fungsi dan hubungan antar *class*

Pada gambar 4 menunjukkan *class diagram* yang menjelaskan *class-class* yang terdapat pada aplikasi surat perjalanan dinas, dapat dilihat bahwasanya antar *class* mempunyai keterkaitan sesuai dengan kebutuhannya, admin harus melakukan login terlebih dahulu untuk bisa masuk ke *dashboard* setelah itu admin bisa melakukan fungsi-fungsi yang mempunyai keterkaitan dengan *dashboard*.



Gambar 4. Class Diagram

#### 4.5. Penerapan *sequential search*

Pencarian berurutan satu per satu pada kumpulan data dengan *keyword* yang akan dicari hingga menemukan data atau tidak menemukan. Pencarian dilakukan dengan melakukan pengulangan data pertama hingga data ke-n. Setiap perulangan akan membandingkan data ke-i dengan *keyword* atau data yang dimaksud dicari. Bila terdapat kesamaan dengan *keyword* yang dicari maka data berhasil ditemukan dan pencarian berhenti. Namun jika tidak terdapat data yang cocok dengan *keyword* yang sedang dicari hingga akhir data maka perulangan akan berhenti dan status data tidak ditemukan.

```
$no = 1;
#foreach untuk pengulangan
foreach ($pegawai as $dt) : ??
<?php if(isset($keyword)): ??
    <!-- Pencarian NIP -->
    <?php if($dt->nip == $keyword): ??
```

Gambar 5. Implementasi Koding *Sequential search*

Gambar 5 merupakan implementasi koding algoritma *sequential search* pada aplikasi surat perjalanan dinas yang diterapkan untuk proses pencarian data pegawai di dinas xyz. Dimana pencarian data akan dilakukan secara berulang dimulai dari data paling awal hingga data paling akhir

```
<?php break;
```

Gambar 6. Penggunaan *Break*

Gambar 6 menjelaskan penggunaan *break* untuk meningkatkan efisiensi proses pencarian. Dengan menggunakan *break*, apabila data yang dicari sudah ditemukan maka pengulangan atau *looping* akan di dihentikan. [12].

#### 4.6. Penerapan *binary search*

Pencarian *binary* melakukan pencarian data dalam data awal hingga akhir dengan dimulai dari data tengah. Namun perlu diingat syarat algoritma *binary search* datanya harus sudah terurut. Prosesnya membagi dua dari total keseluruhan data. Bila data yang dicari lebih kecil dari data tengah, maka pencarian akan berlanjut ke kiri kemudian pencarian akan dilakukan berulang hingga menemukan data yang dicari. Berikut ini merupakan implementasi koding *binary search* :

```
private function binary($pegawai, $cari, $index){
    $pembagian = floor((count($pegawai)-$index)/2);

    if($pegawai[$pembagian]->nip == $cari){
        return $data = [
            'nip' => $pegawai[$pembagian]->nip,
            'nama' => $pegawai[$pembagian]->nama_pegawai
        ];
    }
}
```

Gambar 7.Implementasi Koding *Binary search*

Gambar 7 menjelaskan implementasi algoritma koding *binary search* pada aplikasi surat perjalanan dinas yang diterapkan untuk proses pencarian data pegawai di dinas xyz. Dimana dalam proses pencariannya dimulai dengan membagi seluruh data menjadi dua kemudian melihat dari hasil pembagian apabila hasil pembagian lebih besar daripada data yang dicari maka pencarian akan dimulai dari sebelah kiri untuk menemukan data yang dicari.

#### 4.7. Halaman Login

### LOGIN ADMINISTRATOR

Login

Gambar 8.Halaman *Login*

Gambar 8 merupakan halaman login untuk admin aplikasi surat perjalanan dinas. Yang berhak login ke dalam sistem adalah admin dalam melakukan login ke dalam sistem admin perlu menginput username dan password yang benar agar bisa masuk ke dalam sistem

#### 4.8. Halaman *Dashboard*



Gambar 9.Halaman *Dashboard*

Gambar 9 merupakan halaman *dashboard* aplikasi surat perjalanan dinas dimana pada halaman tersebut menampilkan status jumlah data pegawai dan data surat perjalanan dinas yang terdapat pada aplikasi. Control panel dibuat untuk memudahkan admin untuk mengakses menu-menu yang ada pada aplikasi surat perjalanan dinas.

#### 4.9. Halaman Pencarian

NO	NAMA	NIP	PANGKAT GOLONGAN	JABATAN	AKSI	
1	U	UMR	3	Pembina - I/II	KEPALA DINAS	<a href="#">[i]</a> <a href="#">[x]</a>
2	A. MUK	ANSTW/LST.	3	IV/II	SEKRETARIS	<a href="#">[i]</a> <a href="#">[x]</a>
3	R.	LST, MP, MH	3	IVA	KEPALA SUBBAGIAN PERENCANAAN DAN PELAPORAN	<a href="#">[i]</a> <a href="#">[x]</a>
4	R.	..	3	IVA	KEPALA SUBBAGIAN KEUANGAN DAN ASSET	<a href="#">[i]</a> <a href="#">[x]</a>

Gambar 10.Halaman *Pencarian*

Gambar 10 merupakan halaman pencarian. Pada halaman tersebut di implementasikan 2 metode pencarian yaitu metode *sequential search* dan metode *binary search*. Untuk *keyword* yang di cari adalah NIP pegawai dinas xyz.

#### 4.10. Halaman Hasil Pencarian

Sistem Elektronik Surat Perjalanan Dinas

**Data Pegawai**

[\[x\]](#) [\[i\]](#)

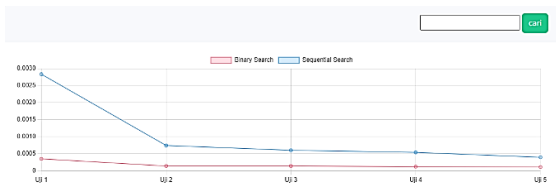
NO	NAMA	NIP	PANGKAT GOLONGAN	JABATAN	AKSI
1	LIANDA	19.....004	IVA	KEPALA LPTD TAMAN 1	<a href="#">[i]</a> <a href="#">[x]</a>

Previous 1 Next

Gambar 11.Halaman *Hasil Pencarian*

Gambar 11 merupakan halaman hasil pencarian data pegawai pada aplikasi surat perjalanan dinas.

4.11. Halaman Pencarian Kebutuhan Proses Perbandingan



Gambar 12. Halaman Pencarian Proses Perbandingan

Gambar 12 merupakan halaman pencarian data pegawai pada aplikasi surat perjalanan dinas dan hasil pencarian kedua metode bisa terlihat pada bentuk grafik.

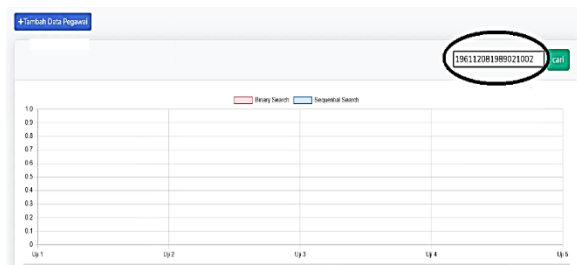
4.12. Pengujian

Pengujian algoritma bertujuan untuk mengetahui perbedaan dari dua algoritma pencarian yang sudah diimplementasikan. Sebelumnya keseluruhan data terdapat 250 data pegawai dan terkait data yang di uji merupakan data yang sudah diurutkan, seperti yang kita tahu bahwa algoritma *sequential search* bisa diterapkan untuk data yang acak atau yang sudah diurutkan sedangkan algoritma *binary search* hanya bisa diterapkan untuk data yang sudah dalam keadaan terurut. Maka pada penelitian algoritma *sequential search* akan mengikuti pola *binary search*. Pengujian kecepatan pencarian dilakukan dengan menggunakan fungsi *microtime*. *Microtime* berfungsi untuk menghitung waktu yang dibutuhkan ketika *page* atau halaman tersebut melakukan proses pencarian. Uji coba dilakukan sebanyak 5 kali untuk mencari data pegawai yang terletak di awal, tengah dan akhir. Setelah melakukan perbandingan maka kita dapat mengetahui algoritma manakah yang lebih baik untuk diterapkan pada aplikasi. Untuk proses pengujian akan dilakukan simulasi di beberapa *device*.

4.13. Proses Pengujian Algoritma

Untuk mendapatkan hasil pengujian perlu dilakukan beberapa tahapan sebagai berikut :

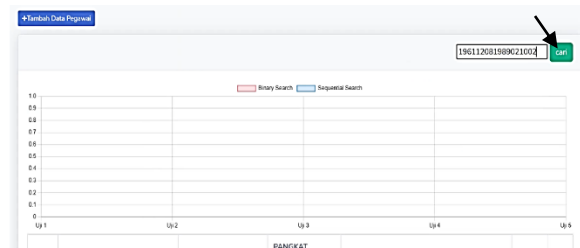
1. Menginput *keyword* yang dicari



Gambar 13. Input Keyword Pencarian

Gambar 13 merupakan proses menginput *keyword* pencarian. *Keyword* yang dicari diatas berupa NIP pegawai.

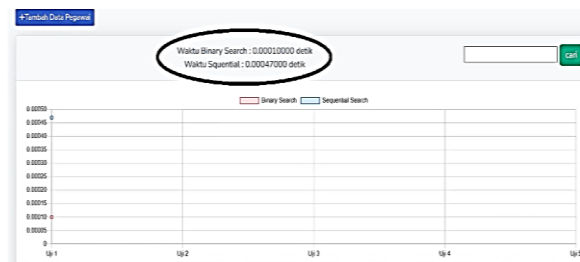
2. Kemudian pilih button cari untuk melakukan proses pencarian data pegawai



Gambar 14. Proses Pencarian

Gambar 14 menunjukkan tanda panah yang menunjuk pada *button* cari untuk melakukan proses pencarian pegawai setelah menginput *keyword* pencarian yang sudah dijelaskan pada tahap 1 sebelumnya.

3. Diketahui hasil waktu perbandingan kecepatan antara algoritma *sequential search* dan algoritma *binary search* seperti yang ditunjukkan pada gambar dibawah ini :



Gambar 15. Hasil Waktu Perbandingan

Gambar 15 menunjukkan hasil setelah sistem melakukan proses pencarian sehingga diketahui hasil waktu perbandingan kecepatan antara algoritma *sequential search* dan algoritma *binary search* dan hasil tersebut digambarkan pada grafik yang dimulai dari sebelah kiri.

4.14. Hasil dan Pembahasan

Dalam pengujian kedua algoritma ini dilakukan simulasi menggunakan 3 *device* yang mana setiap *device* memiliki spesifikasi yang berbeda. Spesifikasi *device* dapat di lihat pada tabel di bawah ini :

Tabel 1. Spesifikasi *Device*

Nama <i>Device</i>	Processor	RAM	Perangkat Penyimpanan
ASUS	Intel core i5	4 GB	Hard Disk 1TB
DELL	Intel core i3	8 GB	Hard Disk 1TB
HP	Intel core i5	4 GB	SSD 256 GB

Setelah melakukan pengujian di beberapa *device* ditemukan hasil kecepatan pencarian dari kedua algoritma, hasil kecepatan pencarian tiap algoritma dapat dilihat pada tabel dibawah ini:

**A. Simulasi Device 1**

Tabel 2. Simulasi Pengujian Device 1

Data Uji	Waktu Pencarian Algoritma	
	Sequential	Binary
Uji 1	0.00108	0.00012
Uji 2	0.00105	0.00013
Uji 3	0.00062	0.00012
Uji 4	0.00080	0.00013
Uji 5	0.0047	0.00011
Rata-rata waktu hasil uji pencarian	0.00165 detik	0,00012 detik

Berikut grafik hasil uji pada device 1 :



Gambar 16. Grafik Hasil Uji Device 1

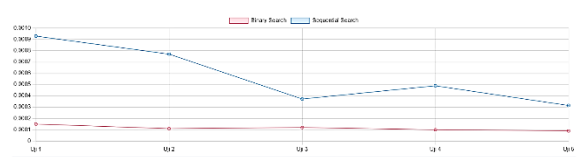
Pada tabel 2 dan grafik 16 adalah hasil uji pada device 1 dengan mengambil 5 data pegawai secara acak yaitu untuk data yang terletak di awal, tengah dan akhir. Pada uji 1 dengan keyword pencarian pada data ke 248 membutuhkan waktu sequential 0.00108 detik dan waktu binary 0.00012 detik. Pada uji 2 dengan keyword pencarian pada data ke 230 membutuhkan waktu sequential 0.00105 detik dan waktu binary 0.00013 detik. Dimana data uji 1 dan uji 2 mewakili pencarian data yang terletak di akhir. selanjutnya pada uji 3 dengan keyword pencarian pada data ke 110 membutuhkan waktu sequential 0.00062 detik dan waktu binary 0.00012 detik. Dimana data uji 3 mewakili pencarian data yang terletak di tengah. Pada uji 4 dengan keyword pencarian pada data ke 12 membutuhkan waktu sequential 0.00080 detik dan waktu binary 0.00013 detik. Pada uji 5 dengan keyword pencarian pada data ke 3 membutuhkan waktu sequential 0.0047 detik dan waktu binary 0.00010 detik. Dimana data uji 4 dan uji 5 mewakili pencarian data yang terletak di awal

**B. Simulasi Device 2**

Tabel 3. Simulasi Pengujian Device 2

Data Uji	Waktu Pencarian Algoritma	
	Sequential	Binary
Uji 1	0.00153	0.00011
Uji 2	0.00059	0.00011
Uji 3	0.00019	0.00010
Uji 4	0.00052	0.00012
Uji 5	0.00045	0.00011
Rata-rata waktu hasil uji pencarian	0.00065 detik	0,00011 detik

Berikut grafik hasil uji pada device 2 :



Gambar 17. Grafik Hasil Uji Pada Device 2

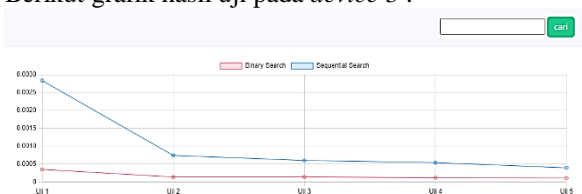
Pada tabel 3 dan grafik 17 adalah hasil uji pada device 2 dengan mengambil 5 data pegawai secara acak yaitu untuk data yang terletak di awal, tengah dan akhir. Pada uji 1 dengan keyword pencarian pada data ke 248 membutuhkan waktu sequential 0.00153 detik dan waktu binary 0.00011 detik. Pada uji 2 dengan keyword pencarian pada data ke 230 membutuhkan waktu sequential 0.00059 detik dan waktu binary 0.00011 detik. Dimana data uji 1 dan uji 2 mewakili pencarian data yang terletak di akhir. selanjutnya pada uji 3 dengan keyword pencarian pada data ke 110 membutuhkan waktu sequential 0.00019 detik dan waktu binary 0.00010 detik. Dimana data uji 3 mewakili pencarian data yang terletak di tengah. Pada uji 4 dengan keyword pencarian pada data ke 12 membutuhkan waktu sequential 0.00052 detik dan waktu binary 0.00012 detik. Pada uji 5 dengan keyword pencarian pada data ke 3 membutuhkan waktu sequential 0.00045 detik dan waktu binary 0.00011 detik. Dimana data uji 4 dan uji 5 mewakili pencarian data yang terletak di awal

**C. Simulasi Device 3**

Tabel 4. Simulasi Pengujian Device 3

Data Uji	Waktu Pencarian Algoritma	
	Sequential	Binary
Uji 1	0.00039	0.00011
Uji 2	0.00054	0.00012
Uji 3	0.0006	0.00014
Uji 4	0.00074	0.00014
Uji 5	0.00283	0.00035
Rata-rata waktu hasil uji pencarian	0.00102 detik	0.00017 detik

Berikut grafik hasil uji pada device 3 :



Gambar 18. Grafik Hasil Uji Device 3

Pada tabel 4 dan grafik 18 adalah hasil uji pada device 3 dengan mengambil 5 data pegawai secara acak yaitu untuk data yang terletak di awal, tengah dan akhir. Pada uji 1 dengan keyword pencarian pada data ke 248 membutuhkan waktu sequential 0.00283 detik dan waktu binary 0.00035 detik. Pada uji 2 dengan



keyword pencarian pada data ke 230 membutuhkan waktu *sequential* 0.00074 detik dan waktu *binary* 0.00014 detik. Dimana data uji 1 dan uji 2 mewakili pencarian data yang terletak di akhir. selanjutnya pada uji 3 dengan *keyword* pencarian pada data ke 110 membutuhkan waktu *sequential* 0.00060 detik dan waktu *binary* 0.00014 detik. Dimana data uji 3 mewakili pencarian data yang terletak di tengah. Pada uji 4 dengan *keyword* pencarian pada data ke 12 membutuhkan waktu *sequential* 0.00054 detik dan waktu *binary* 0.00012 detik. Pada uji 5 dengan *keyword* pencarian pada data ke 3 membutuhkan waktu *sequential* 0.00039 detik dan waktu *binary* 0.00011 detik. Dimana data uji 4 dan uji 5 mewakili pencarian data yang terletak di awal

\*) Catatan

ketika melakukan proses pencarian data bersamaan dengan menampilkan halaman / template (*html*) akan mengalami proses penampilan halaman secara utuh maka hasil bisa saja berbeda-beda sesuai dengan kondisi aplikasi browser setiap *device*

Setelah melakukan beberapa simulasi dari beberapa *device* dengan masing-masing spesifikasi yang berbeda di dapatkan hasil rata-rata waktu pencarian pada tiap algoritma, penulis membandingkan kedua hasil rata-rata waktu pencarian yang didapatkan dalam bentuk tabel sebagai berikut.

Tabel 5. Hasil Rata-Rata Waktu Pencarian

Algoritma	Rata-rata waktu pencarian		
	Device 1	Device 2	Device 3
<i>Sequential</i>	0.00165 detik	0.00065 detik	0.00102 detik
<i>Binary</i>	0.00012 detik	0.00011 detik	0.00017 detik

Berdasarkan hasil rata-rata waktu pencarian dapat dilihat pada tabel 5 bahwa terdapat perbedaan yang cukup signifikan pada kedua algoritma pencarian berdasarkan waktu kecepatan pencarian. Dari simulasi ketiga *device* dapat dilihat juga bahwa algoritma *sequential search* membutuhkan waktu yang lebih lama dibandingkan algoritma *binary search*. Sehingga bisa dikatakan bahwa algoritma *binary search* lebih baik dalam melakukan proses pencarian data berdasarkan waktu kecepatan pencariannya.

## 5. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Setelah melakukan analisis dan implementasi pada algoritma *sequential search* dan *binary search* pada pencarian data pegawai berdasarkan parameter waktu kecepatan pencariannya dapat disimpulkan bahwa algoritma *binary search* memiliki performa kecepatan pencarian yang lebih baik dibandingkan dengan algoritma *sequential search*. Algoritma *binary search* cenderung lebih stabil dan membutuhkan waktu yang sedikit dalam melakukan proses pencarian

data. Sedangkan algoritma *sequential search* membutuhkan waktu yang lebih lama dibandingkan algoritma *binary search*, *sequential search* dapat mencari data dengan cepat apabila data yang dicari terletak di awal, tetapi apabila data yang dicari terletak di akhir kecepatan pencarian akan semakin lama. sehingga *binary search* akan lebih cocok untuk diterapkan pada aplikasi surat perjalanan dinas atau pencarian data pada aplikasi dengan jumlah data yang banyak atau besar.

### 5.2. Saran

Untuk penelitian selanjutnya disarankan dapat menambah parameter pengujian atau menganalisis algoritma pencarian lain yang memiliki pola yang lebih mirip dengan *binary search* untuk mengetahui adakah algoritma yang lebih baik performanya sehingga mendapatkan hasil yang lebih maksimal.

## DAFTAR PUSTAKA

- [1] I. Handayani, E. Febriyanto, and M. Shofwatullah, "Optimalisasi visibilitas situs iLearning Journal Center (iJC) Pada Mesin Pencari Berbasis Search Engine Optimization (SEO) On Page," *SATIN - Sains dan Teknol. Inf.*, vol. 5, no. 1, pp. 27–35, 2019, doi: 10.33372/stn.v5i1.448.
- [2] Akmam and H. Amir, "Pengaruh Pembelajaran Generatif Berbasis Strategi Konflik Kognitif Terhadap Kompetensi Mahasiswa Dalam Mata Kuliah Algoritma Dan Pemrograman Komputer," 2546.
- [3] G. G. Maulana, "Pembelajaran Dasar Algoritma Dan Pemrograman Menggunakan El-Goritma Berbasis Web," *J. Tek. Mesin*, vol. 6, no. 2, p. 8, 2017, doi: 10.22441/jtm.v6i2.1183.
- [4] M. Rizky, "Pembelajaran Dasar-Dasar Algoritma Dan Pemrograman," 2546.
- [5] I. Nurul, "Perbandingan Algoritma *Sequential search* Dan Algoritma *Binary search* Pada Aplikasi Kamus Bahasa Indonesia Menggunakan Php Dan JQuery," *Comput. J. Inform.*, vol. 08, pp. 1–6, 2021, [Online]. Available: <https://unibba.ac.id/ejournal/index.php/computing/article/view/564%0Ahttps://unibba.ac.id/ejournal/index.php/computing/article/download/564/468>.
- [6] Nurwanto, "Aplikasi Perpustakaan Berbasis Web," vol. 13, no. 1, pp. 24–31, 2011.
- [7] R. D. A. Aviantika, K. Kustanto, and M. Hasbi, "Pencarian Data Barang Produk Atribut Sekolah Menggunakan Algoritma *Binary search*," *J. Teknol. Inf. dan Komun.*, vol. 9, no. 1, p. 75, 2021, doi: 10.30646/tikomsin.v9i1.546.
- [8] Deva A.R.S., "Penerapan Algoritma *Binary search* Pada Aplikasi E-Order (Studi Kasus Paris Van Java Kota Bengkulu)," *Jukomika*, vol. 3, no. 6, pp. 571–580, 2020, [Online]. Available: <https://jurnal.ikhafi.or.id/index.php/jukomika/571>.

- [9] B. Sulistio *et al.*, “Aplikasi Kamus Bahasa Taliabu Berbasis Android dengan Menggunakan Metode *Binary search*,” vol. 2, no. 2, pp. 67–72, 2019.
- [10] Y. Rahmanto, J. Alfian, D. Damayanti, and R. I. Borman, “Penerapan Algoritma *Sequential search* pada Aplikasi Kamus Bahasa Ilmiah Tumbuhan,” *J. Buana Inform.*, vol. 12, no. 1, p. 21, 2021, doi: 10.24002/jbi.v12i1.4367.
- [11] P. Muhammad, Akbar, “Analisis Perbandingan Algoritma *Linear* dan *Binary* pada Aplikasi Kamus Bahasa Indonesia-Jawa Skripsi Oleh Muhammad Akbar Pratama Analisis Perbandingan Algoritma *Linear* dan *Binary* pada Aplikasi Kamus Bahasa Indonesia-Jawa Skripsi Oleh Muhammad Akbar Pratama,” 2020.
- [12] R. Yoga, “Analisis Algoritma *Sequential search* Dan *Binary search* Pada Big Data,” vol. 14, no. 1, pp. 74–79, 2019.
- [13] R. Aditya, V. H. Pranatawijaya, and P. B. A. A. Putra, “Rancang Bangun Aplikasi Monitoring Kegiatan Menggunakan Metode Prototype,” *J. Inf. Technol. Comput. Sci.*, vol. 1, no. 1, pp. 47–57, 2021, [Online]. Available: <https://e-journal.upr.ac.id/index.php/jcoms/article/view/2955>.
- [14] A. Ni Made, “Analisa dan Perancangan Aplikasi Pembelajaran Bahasa Inggris Dasar Berbasis Android,” *J. IKRAITH-INFORMATIKA*, vol. 1, no. 3, pp. 107–115, 2020.
- [15] G. Saputri, “Perancangan Sistem Informasi Rincian Biaya Produk Berbasis Web Pada Pt . Abc,” *J. Teknol. Inf. ESIT*, vol. XV, no. 01, pp. 41–48, 2020.