Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 294-306
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Kombinasi Algoritma *Sequential Searching* dan *Bubble Sort* Pada Manajemen Laboratorium

Nur Arifin[1], Fauziah[2], Nurhayati[3]
[1,2,3]Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional
Email: [1]arifingdr@gmail.com, [2]Fauziah@civitas.ac.id, [3]Nurhayati@civitas.ac.id

Abstract

The facilities in the laboratory can make students especially feel comfortable because of several supports such as internet availability, better computer specifications, servers with large capacities as data storage media and rooms that support students' conditions in doing their assignments. During the pandemic, the laboratory cannot be used optimally because learning is done online. So that an online application for borrowing a laboratory is needed to maximize the function of the laboratory. In this study, a laboratory loan website application was designed by combining sequential search and bubble sort algorithms on the search engine menu so as to facilitate and speed up the process of searching for laboratory types. The test was carried out simultaneously from 700 test data and resulted in a time of 3 milliseconds for the sequential search algorithm, and 0.036 milliseconds for the bubble sort algorithm. While the results of the testing time with the combination of the two algorithms obtained are 4 milliseconds.

Keywords: *Laboratories, Sequential Search, Bubble Sort*

Abstrak

Sarana pada laboratorium dapat membuat para mahasiswa merasa nyaman karena beberapa penunjang seperti ketersediaan internet, spesifikasi komputer yang lebih baik, server dengan kapasitas yang besar sebagai media penyimpanan data dan ruangan yang menunjang kondisi mahasiswa dalam mengerjakan tugasnya. Pada masa pandemi sarana, laboratorium tidak dapat digunakan secara maksimal dikarenakan pembelajaran dilakukan secara online. Sehingga diperlukan aplikasi online peminjaman labratorium untuk memaksimalkan fungsi dari laboratorium. Pada penelitian ini dirancang sebuah aplikasi website peminjaman laboratorium dengan mengkombinasikan algoritma sequential search dan bubble sort pada menu search engine sehingga memudahkan dan mempercepat proses pencarian jenis laboratorium. Pengujian dilakukan secara serentak dari 700 data uji dan menghasilkan waktu sebesar 3 milliseconds untuk algoritma sequential search, dan 0.036 milliseconds untuk algoritma bubble sort. Sedangkan hasil waktu pengukian dengan kombinasi kedua algoritma didapatkan sebesar 4 milliseconds.

Kata kunci: *Laboratorium, Sequential Search, Bubble Sort*

1. PENDAHULUAN

Laboratorium adalah ruangan atau ruang tempat dilaksanakannya kegiatan praktikum atau penelitian yang didukung oleh seperangkat alat dan prasarana laboratorium yang lengkap[1]. Pembelajaran praktik secara tidak langsung memiliki tujuan yang baik untuk mahasiswa agar mampu menguasai materi pembelajaran yang lebih baik. Menurut Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Republik Indonesia pada tahun 2010 No.3 bersama Menteri Pendidikan Nasional dan Badan Kepegawaian Negara tahun 2010 No.2, menjelaskan "Laboratorium bersifat tetap atau bergerak, dikendalikan secara sistematis untuk kegiatan pengujian, kalibrasi, dan diproduksi sampai batas tertentu, dengan

menggunakan peralatan dan bahan berbasis ilmu pengetahuan tertentu Unit penunjang keilmuan bagi lembaga pendidikan berupa ruang tertutup atau ruang terbuka Metode dalam rangka penyelenggaraan pendidikan, penelitian, dan pengabdian kepada masyarakat"[2]. Sistem manajemen laboratorium yang sistematis dan efektif. Salah satunya menggunakan web dinamis yang terorganisir dalam database. Ini sering disebut sebagai sistem manajemen inspeksi[3].

Pada penelitian *sorting* sebelumnya, Saat aplikasi dirancang menggunakan algoritma *bubble sort*, aplikasi diuji dengan 200 data, memberikan peringkat rendah hingga sedang, dan mencapai skor kepatuhan 100% pada keempat aspek yang diukur[4]. Namun ada juga yang melakukan perbandingan pengurutan kecepatan dan melakukan gabungan 3 algoritma *quick sort*, *merge sort, bubble sort* namun pada hasil yang dibandingkan pada pembahasan ini menyimpulkan algoritma *quick sort* adalah algoritma tercepat dari ke 3 algoritma yang sudah disebutkan[5]. Pada penelitian *searching* sebelumnya, tidak menjelaskan hasil secara rinci bagaimana hasil algoritma *searching*-nya, namun dapat disampaikan pada penelitian sebelumnya hanya mampu menjelaskan bahwa aplikasi berjalan menggunakan algoritma *searching* pada tampilan data pencarian[6]. Sesuai dengan penulisan A.S. Gowtham proses perbandingan mendefinisikan tempat untuk melihat transformasi angka, perbandingan dilakukan berpasangan, transformasi angka meningkat pada waktu penyortiran array[7].

Pencarian aplikasi laboratorium ini dilakukan dengan menerapkan *sequential search*, yaitu tindakan yang mengambil data dari dalam kumpulan *dataset*. *Sequential search* merupakan metode pencarian data yang paling sederhana, dan pencarian data ini dilakukan dengan membandingkan array yang ada[8]. Pengurutan *bubble sort* dapat didasarkan nilai maksimum untuk setiap array yang menukar komponen satu demi satu. *bubble sort* pada dasarnya membawa kompleksitas dan juga merupakan algoritma yang paling sederhana, tetapi sangat kuat untuk dibandingkan dengan algoritma lainnya. Dan ini dapat digunakan untuk menguruti angka N dalam urutan menaik atau menurun. Pada dasarnya, *bubble sort* dilakukan dalam 3 langkah. Pertama mencoba meneria semua input dari awal hingga akhir, kedua mengurutkan gelembung yang dipasang sesuai dengan urutan data, Ketiga membandingkan setiap urutan array kecil atau besarnya angka[9].

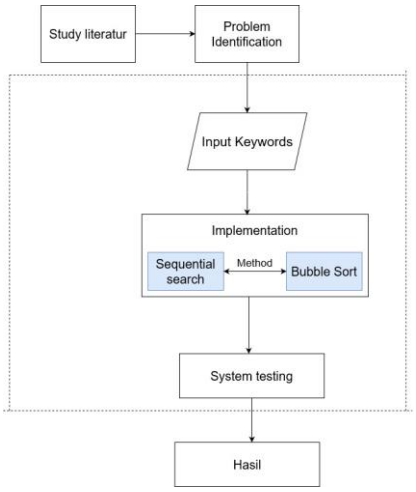
Pada penelitian ini menjelaskan mengenai eksperimen komparatif berdasarkan evaluasi waktu *sequential search*, *bubble sort* dan kombinasi *sequential search* dan *bubble sort* sistem implementasi yang digunakan adalah bahasa pemrograman PHP, JavaScript. Hasil waktu berlangsung ketika *input* pencarian sedang dilakukan, untuk menghasilkan waktu dengan satuan *miliseconds*.

2. METODOLOGI PENELITIAN

Penelitian ini yang berkaitan dengan manajemen laboratorium di Universitas Nasional. Menggunakan teknik pencarian dan sortir Pada kondisi

saat ini menyewa laboratorium pada permasalahan waktu. Akhirnya, para peneliti menemukan ide untuk mengembangkan metode algoritma yang ada untuk menyelesaikan masalah ini. Penelitian observasi ini telah melibatkan sebagian mahasiswa di universitas nasional.

Metode pada penelitian ini secara sistematis dibagi menjadi beberapa tahapan yang terdiri dari pengumpulan data, metode(*sequential search* dan *bubble sort*), *system testing* dan *deployment* yang bertujuan untuk memperoleh informasi yang dibutuhkan dalam mencapai hasil penelitian[10], yang dijelaskan pada Gambar 1:



Gambar 1. Tahapan penelitian

Gambar 1 menjelaskan tahapan metode penelitian, menggunakan algoritma *sequential search* dan *bubble sort* sebagai berikut :

a) Study literatur memberikan pengetahuan atau wawasan dengan upaya melanjutkan penelitian yang masih perlu di teliti lebih lanjut.
b) Setelah membaca beberapa study literatur dan permasalahan ditemukan peneliti mencoba lebih lanjut mencari tahu masalah pencarian menggunakan *sequential search* dan *bubble sort*
c) Lalu mengimplementasi kedua metode tersebut di pencarian laboratorium
d) Untuk mendapatkan hasil yang optimal penelitian melakukan perobaan dengan 700 data.

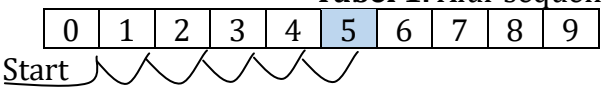
2.1. Algoritma *Sequential Searching*

Sequential searching (juga dikenal sebagai pencarian linier) memiliki model pencarian paling sederhana yang dapat dilakukan pada kumpulan data. Secara konseptual dapat dijelaskan sebagai berikut. Terdapat sebuah data untuk variabel M. Yang memiliki sebuah array yang berisi *n* data (M[0], M[1],...,M[*n*-1], dan K adalah data berikut, yang dicari) M[i] = K Pencarian dilakukan untuk menemukan dia. Dimana i adalah bilangan indeks terkecil yang memenuhi $0 \leq k \leq n$-1[11].

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 294-306
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Proses pengambilan data menggunakan metode ini sangat mudah. Dengan menyusun dapat dilakukan dengan menyusun data secara berurutan, dari data awal hingga data paling akhir. Namun jika hasil yang diambil sama, data yang cocok berhasil ditampilkan, tetapi jika datanya tidak sama, nilai kosong dikembalikan[12].

Tabel 1. Alur sequential search.



Tabel 1 adalah Proses *sequential searching* dimulai dengan elemen pertama dan membandingkan setiap elemen array satu demi satu hingga elemen yang dicari ditemukan dan semua elemen diperiksa sebagai berikut:

M[i] = k

Dimana i adalah angka indeks terkecil yang memenuhi syarat $0 \leq k \leq n-1$ Tentu saja, di mungkin tidak menemukan data yang dicari. Contoh:

M ←[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Untuk mencari angka 5 dalam hal ini terhitung pada indeks 5[13].

Berikut cara kerja dari *sequential search* [14]:

1. i ← 0
2. ditemukan ← false
3. selama (tidak ditemukan) dan (I <= n) menuju kelangkah 4
4. jika (M[i] = x) maka ditemukan ← true, jika tidak i← i + 1
5. jika ditemukan maka indek itu yang dicari,jika tidak ada data tidak ditemukan.

2.2. Algoritma *Bubble Sort*

Algoritma *bubble sort* merupakan salah satu pengurutan yang sangat sederhana, algoritma *bubble sort* melakukan perbandingan semua data satu per satu kepada data disampingnya yang ada didalam daftar[15]. Prinsip kerja dari *bubble sort* ini adalah *ascending* dimana maksimum diurutkan ke minimum. Dinilai berdasarkan skor yang dicapai oleh mahasiswa[10].

Laboratorium yang akan diperlihatkan dengan kiteria penelitian ini menerapkan algoritma *bubble sort*. Dibawah ini merupakan langkah pengurutan algoritma *bubble sort* [4]:

a) Membandingkan Z[1] kepada Z[2], kemudian dikemas lagi berdasarkan alur dan disesuaikan menjadi Z[1] < Z[2].
b) Membandingkan Z[2] dan Z[n] lagi, kemudian mengurutkan data yang disesuaikan menjadi Z[2] < Z[n].
c) Membandingkan Z[n1] dan Z[n] kemudian mengurutkan data yang disesuaikan menjadi Z[n-1] < Z[n], Sesudah membandingkan Z[n] dan (n1), maka Z[n] adalah element terurut terbesar pertama.

Namun Algoritma *bubble sort* ini sangat mudah diimplementasikan, dan algoritma ini memiliki kompleksitas waktu yaitu $O(n^2)$[16]. Prinsip dasar dari bubble sort adalah membandingkan dua indeks secara berurutan dari kiri ke

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 294-306
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

kanan dan menukarnya ketika kondisi terpenuhi. Pada daftar L = {5, 2, 3, 1} digunakan untuk pengurutan menaik. Dalam contoh ini, setiap pertukaran data yang berhasil menggeser jumlah maksimum ke tempat yang tepat[17].

5	2	3	1

Tahap 1 – Percobaan 1

2	5	3	1

Tahap 2 – Percobaan 1

2	3	5	1

Tahap *3* – Percobaan 1

2	3	1	5

Tahap 1 – Percobaan 2

2	3	1	5

Tahap 2 – Percobaan 2

2	1	3	5

Tahap 3 – Percobaan 2

2	1	3	5

Tahap 1 – Percobaan 3

1	2	3	5

Tahap 2 – percobaan 3

1	2	3	5

Tahap 3 - Percobaan 3

Hasil Akhir

1	2	3	5

Berikut perhitungan jumlah perbandingan maksimal selama ini pada pengurutan metode *bubble sort*:

$$(n - 1) + (n - 2) + (n - 3) + \ldots + 2 + 1 = \sum_{i=1}^{n-1} i = \frac{n\,(n-1)}{2}$$

Efisiensi ini berlaku dalam kasus terbaik, terburuk, dan rata-rata. Jika n = 5, jumlah perbandingan data adalah ((5x4) / 2) = 10 kali. Oleh karena itu, ketika dinyatakan sebagai Big O, kompleksitas *bubble sort* sama dengan O (n (n1) / 2) atau O (n^2). Kasus terbaik adalah ketika data yang diurutkan sudah dalam keadaan terurut[18].

3. HASIL DAN PEMBAHASAN

3.1. Data Laboratorium

Saat ini ada sekitar 6 laboratorium saja yang sudah di inputkan di dalam aplikasi sebagai penentu keywords yang akan dicari oleh mahasiswa yang ingin melakukan pemesanan, namun ada beberapa laboratorium lagi yang belum di inputkan karena situasi pandemi yang sulit mendapatkan akses ke kampus sehingga hanya 6 laboratorium yang bisa didapatkan untuk melakukan penelitian ini. Namun tidak menutup kemungkinan bahwa

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 294-306
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

laboratorium ini bisa di lakukan penambahan oleh admin jika memang laboratorium tersedia.

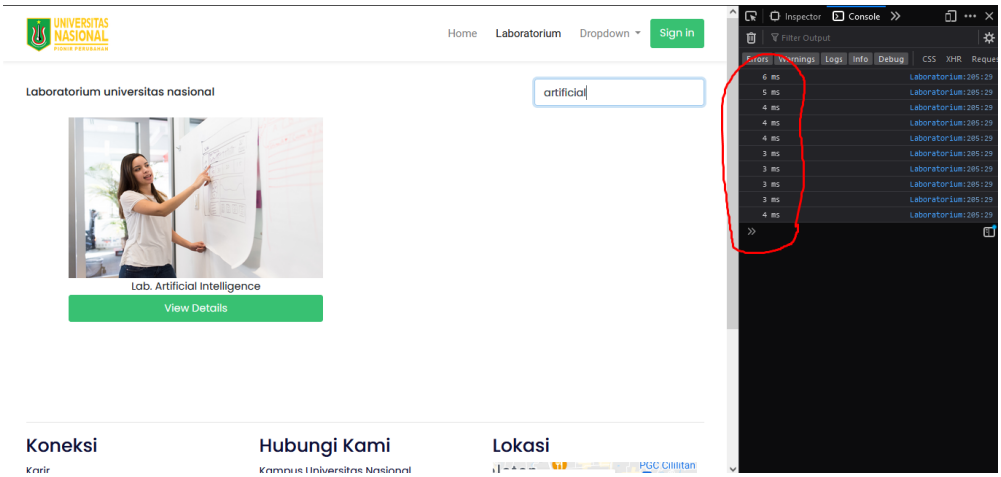
3.2. Implementasi Algoritma *Sequential Search* pada proses pencarian keyword

Sistem implementasi pencarian ini terjadi secara *real-time* atau dilakukan secara langsung di setiap inputan *keyword* yang ada, nantinya mahasiswa mendapatkan hasil *response* waktu secara langsung di dalam *console.log.* Berikut tampilan source code pada metode ini Gambar 2:

```
public function whereAllLab(Request $request)
{
    $data =  DetailRuangan::with(['Gallery'])->where('title', 'LIKE', "%$request->keywords%")->get();
    return response()->json($data);
}
```

Gambar 2. Source code *sequential search*

Hasil pada kode program memberikan *response* waktu yang berupa *miliseconds* yang terus tampil disetiap mahasiswa menginputkan *keyword* pencarian, pada Gambar 3:



Gambar 3. Hasil waktu *sequential search*

Gambar 3 menunjukan pencarian *sequential search* dan memperlihatkan waktu disetiap inputan pencarian. Terdapat hasil berbeda-beda disetiap pencarian yang menunjukan kompleksitas waktu, serta mencocokan data disetiap indexnya hingga memenuhi syarat pada data yang dicari.

3.3. Implementasi Algoritma *bubble sort* pada proses *sorting*

Implementasi algoritma *bubble sort* terjadi ketika dalam proses halaman laboratorium dikunjungi, dan ada hasil waktu yang diperoleh setelah tampilan selesai dirender. Dalam implmentasi ini memiliki 6 laboratorium yang disorting sesuai huruf abjad secara burutuan.

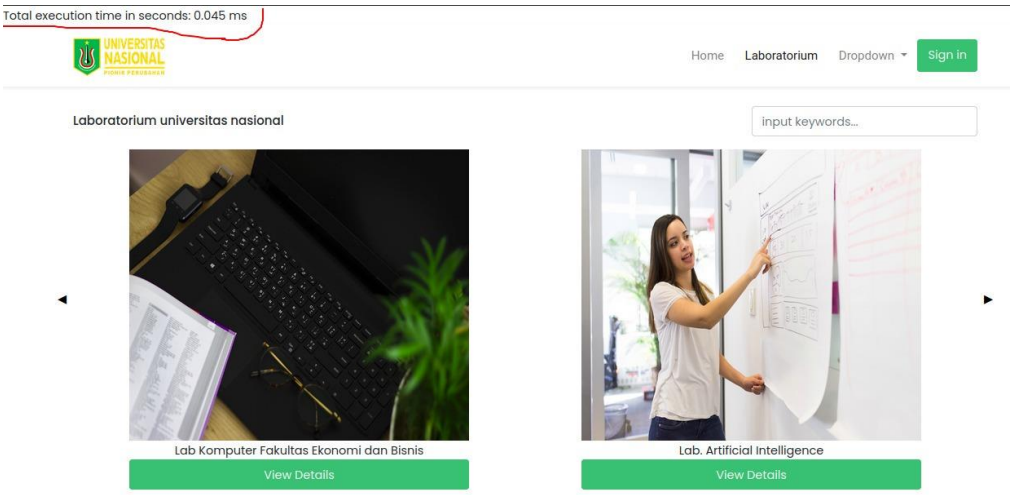
Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 294-306
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

```php
$time_start = microtime(true);
$lab =  DetailRuangan::with(['Gallery'])->orderBy('title', 'ASC')->get();
echo 'Total execution time in seconds: ' . round((microtime(true) - $time_start), 3). ' ms';

// $searchLive = DetailRuangan::with(['Gallery'])->where('title' , 'LIKE', $request->keywords)->get();
return view('pages.user.Laboratorium',[
    'lab'           => $lab,
    // 'searchLive'    => $searchLive
]);
```

Gambar 4. *Source code bubble sort*

Pada Gambar 4 *bubble sort* dihitung menggunakan *microtime*. Dengan melakukan masuk halaman yang berarti waktu dimulai, lalu mengambil data dan mengurutkan data secara *ascending* pada abjad *title*, setelah pengambilan data selesai akan menampilkan kecepatan waktu yang berupa *seconds*, dengan perhitungan waktu selesai dikurang waktu mulai, supaya menampilkan miliseconds kita bisa menggunakan metode round() PHP yang menjadikan tiga angka dibelakang koma, berikut hasil waktu pada Gambar 5.



Gambar 5. Hasil perhitungan *bubble sort*

Perhitungan pada Gambar 5 menunjukan hasil 0.045 *milliseconds* dengan asumsi semua data diurutkan sesuai iterasi abjad terdepan dan dibandingkan satu persatu.

3.4. Kombinasi Algoritma *Sequential Search* Algoritma *bubble sort*

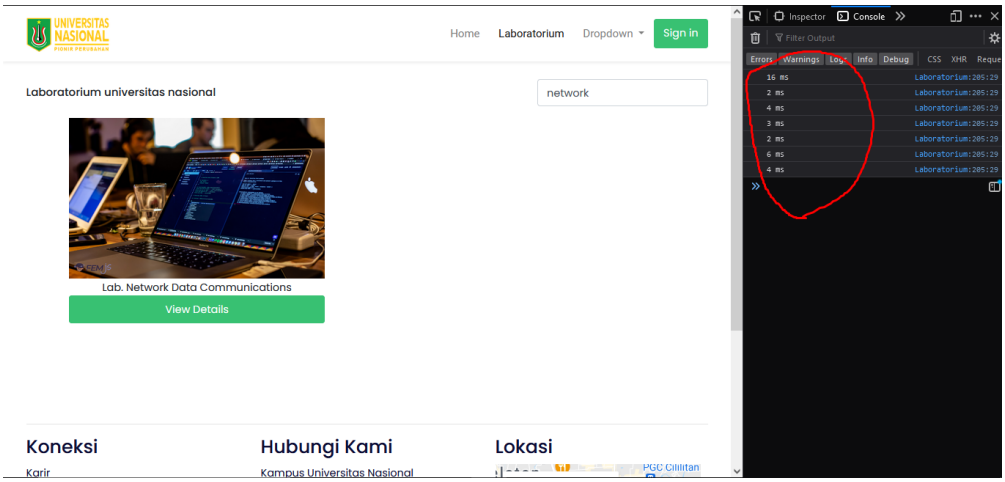
Pencarian ini terjadi secara *real-time* atau dilakukan secara langsung disetiap inputan *keyword* yang ada, namun perbedaanya akan menambahkan kompleksitas pada *script* yang mengharuskan proses pengurutan data dan pencarian data secara *ascending* berupa hasil *response* waktu yang ditampilkan di *console.log.* Berikut tampilan source code Gambar 5:

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 294-306
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

```
public function whereAllLab(Request $request)
{
    $data =  DetailRuangan::with(['Gallery'])->where('title', 'LIKE', "%$request->keywords%")->orderBy('id', 'ASC')->get();
    return response()->json($data);
}
```

Gambar 6. *Source code* kombinasi *sequential search* dan *bubble sort*

Hasil pada *source code* Gambar 6 memberikan hasil *response* waktu yang berupa *miliseconds* yang terus tampil disetiap mahasiswa menginputkan *keyword* pencarian, ini memungkinkan pencarian laboratorium lebih lama di bandingkan implementasi sebelumnya.



Gambar 7. *sequential search* menggunakan *bubble sort*

Gambar 7 menunjukan pencarian menggunakan *bubble sort* dan memperlihatkan waktu disetiap inputan pencarian. Terdapat hasil yang berbeda-beda juga disetiap pencarian yang menunjukan bahwa kompleksitas terhadap kombinasi waktu rumus algoritma *sequential search* dan algoritma *bubble sort*, hal tersebut mencocokan data terlebih dahulu disetiap indexnya hingga memenuhi syarat pada data yang dicari, lalu menghitung kompleksitas waktu terhadap *bubble sort* dengan upaya data yang dicari bisa mengembalikan data secara berurutan.

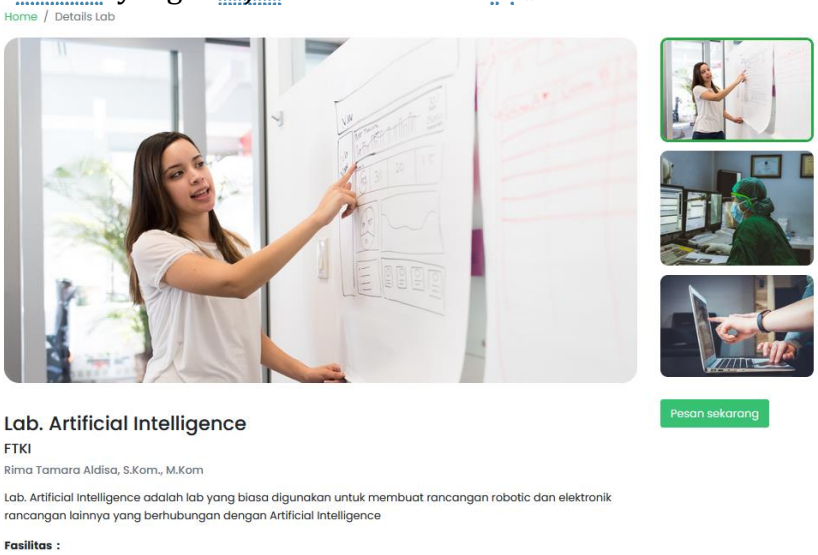
3.5. Implementasi Transaksi

Sistem transaksi pemesanan dapat dilakukan ketika mahasiswa mencari sebuah laboratorium yang cocok dan tentunya sesuai dengan fakultas. Gambar 6 memiliki pesan gagal ketika meminjam pada fakultas lain.

Gambar 8. Gagal pemesanan laboratorium

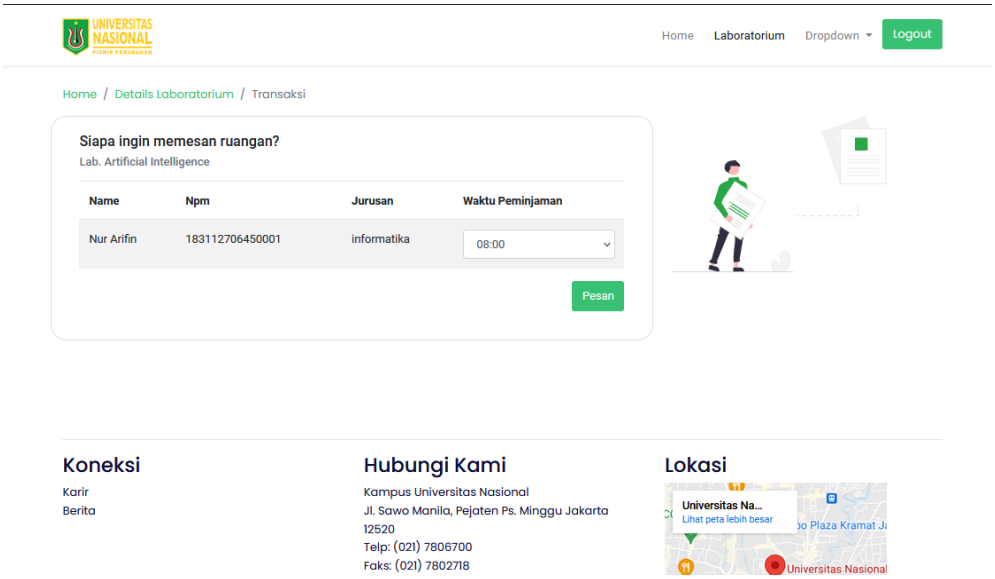
Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 294-306
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Gambar 8 memberikan pesan error ketika proses pemesanan tidak sesuai dengan fakultas yang dituju.



Gambar 9. Laboratorium deskripsi

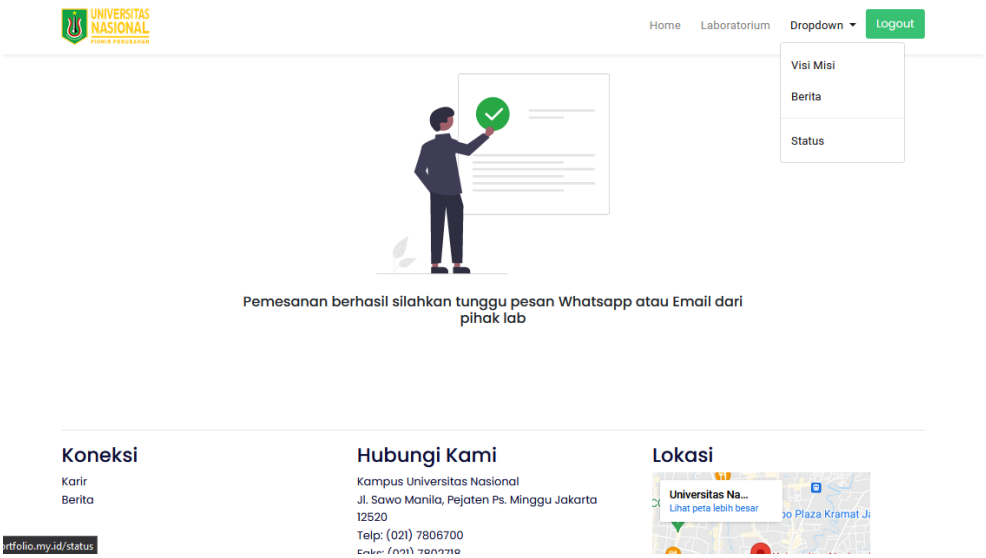
Namun ketika mahasiswa berhasil memesan laboratorium langkah selanjutnya mahasiswa akan diperlihatkan tampilan laboratoriumnya dengan deskripsi yang cukup lengkap supaya mahasiswa paham kegunaan laboratorium, pada Gambar 9 adalah tampilan deskripsi mengenai laboratorium yang dipilih mahasiswa dapat langsung memesan.



Gambar 10. Transaksi pemesanan

Gambar 10 mahasiswa hanya memilih waktu dan melihat data mahasiswa yang memesan waktu peminjaman hanya di pukul 08:00 sampai dengan pukul 20:00 waktu laboratorium ini menyesuaikan jam operasional kampus.

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 294-306
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti



Gambar 11. Pemesanan berhasil

Gambar 11 menunjukan keberhasilan pemesanan laboratorium yang sudah dipesan mahasiswa, langkah selanjutnya mahasiswa dapat melihat status pada peminjaman ini di navbar dropdown status.

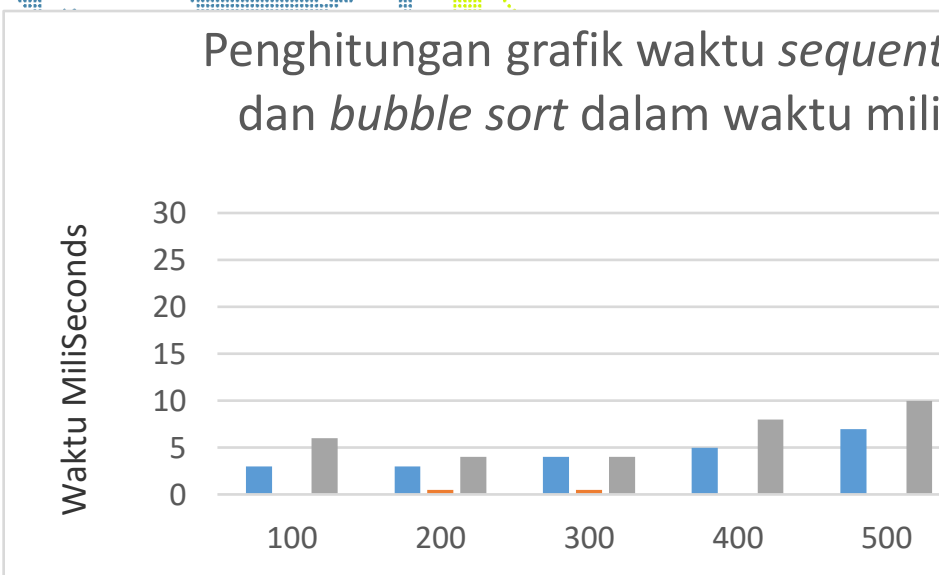
3.5. Hasil

Hasil pengujian ini berdasarkan 700 data yang dengan input pencarian. Berikut hasil waktu eksekusi pencarian *sequential search, bubble sort* dan kombinasi *Sequential search* dan *bubble sort.*

Tabel 2. Waktu pencarian sequential search menggunakan bubble sort dan tidak menggunakan bubble sort

N	***Sequential search***	***Bubble sort***	**Kombinasi *Sequential search* dan *bubble sort***
100	3 *milliseconds*	0.036 *milliseconds*	6 *milliseconds*
200	3 *milliseconds*	0.051 *milliseconds*	4 *milliseconds*
300	4 *milliseconds*	0.052 *milliseconds*	4 *milliseconds*
400	5 *milliseconds*	0.047 *milliseconds*	8 *milliseconds*
500	7 *milliseconds*	0.049 *milliseconds*	10 *milliseconds*
600	8 *milliseconds*	0.079 *milliseconds*	10 *milliseconds*
700	11 *milliseconds*	0.058 *milliseconds*	25 *milliseconds*

Hasil pada Tabel 2 menunjukan hasil kecepatan pada pencarian antara algoritma *sequential search* dan *bubble sort* serta kombinasi kedua algoritma tersebut. Dari 700 data uji, semuanya memiliki waktu yang berbeda-beda disetiap range data ujinya dan dari hasil ini dapat disimpulkan bahwa aplikasi dapat mempermudah pencarian dengan huruf abjad yang tersusun tanpa harus melakukan kembali proses pencarian satu-satu terhadap data lainnya



Gambar 12. Grafik waktu pencarian

Gambar 12 waktu eksekusi *sequential searching* dan *bubble sort* memiliki informasi dalam grafik menunjukan bahwa mencari laboratorium secara bersamaan dengan kombinasi terus meningkat seiring bertambahnya jumlah data yang mengikuti pencarian. Proses penginputan pencarian dengan *bubble sort* membutuhkan waktu cukup lama dibandingkan tidak menggunakan *bubble sort* yang diharuskan menghitung terlebih dahulu dalam pengurutan array, sehingga dibutuhkan penelitian lebih lanjut.

4. SIMPULAN

Berdasarkan hasil pembahasan di atas, hasil yang diperloeh dari 700 data diuji terhadap algoritma *sequential searching* yang diterapkan pada proses pencarian memiliki kecepatan waktu sebesar 3 *milliseconds* dan *bubble sort* pada proses sorting memiliki kecepatan sebesar 0.036 *milliseconds. S*edangkan hasil kombinasi algoritma *sequential search* dan *bubble sort* memiliki waktu sebesar 4 *milliseconds*

DAFTAR PUSTAKA

[1] T. R. Reno Sari, "Aplikasi Sistem Informasi Dan Manajemen Laboratorium," 2017, p. 212.

[2] D. Meidelfi and T. Lestari, "Aplikasi Peminjaman Laboratorium Pada Fakultas Teknologi Pertanian Universitas Andalas," vol. 2, no. 2, pp. 42–47, 2021.

[3] S. Suryaningsih, "Pengembangan Sistem Pengelolaan Administrasi Laboratorium Fisika UIN Walisongo Semarang Berbasis WEB," pp. 1–11, 2017.

[4] H. Pane, Fauziah, and Nurhayati, "Rancang Bangun Aplikasi Kraepelin Test Berbasis Web Menggunakan Metode Bubble Sort," *JOINTECS (Journal Inf. Technol. Comput. Sci.*, vol. 7, no. 1, pp. 41–48, 2018.

[5] A. Sonita and F. Nurtaneo, "Analisis Perbandingan Algoritma Bubble Sort, Merge Sort, Dan Quick Sort Dalam Proses Pengurutan Kombinasi Angka Dan Huruf," *Pseudocode*, vol. 2, no. 2, pp. 75–80, 2016, doi: 10.33369/pseudocode.2.2.75-80.

[6] A. H. M. Susanto Susanto, "Management System Fertilizer Ship Arrival At UPP Semarang Based Website Using Sequential Searching Method," *IJCCS(Indonesian J. Comput. Cybern. Syst.*, vol. 15, no. 4, pp. 1–11, 2021, doi: 10.22146/ijccs.68204.

[7] A. S. Gowtham, "one-dimensional array using ram based sorting algorithm," vol. 7, no. 15, pp. 5904–5914, 2020.

[8] M. H. M. Tonny, ibnu rasyid munthe, "Perancangan Aplikasi Pengenalan Tokoh Penemu Benda-benda Penting di Dunia Berbasis Android Menggunakan Metode Sequential Search," *MEANS (Media Inf. Anal. dan Sist.*, vol. 6, no. 1, pp. 90–94, 2021.

[9] A. F. Dwi M J Purnomo, Ahmad Arinaldi, Dwi T Priyantini, Ari Wibisono, "Implementation Of Serial And Parallel Bubble Sort On Fpga," *J. Ilmu Komput. dan Inf.*, vol. 9, no. 2, p. 113, 2016.

[10] H. Triansyah, "Implemetasi Metode Bubble Sort Dalam Pengurutan Indeks Prestasi Mahasiswa," *J. Ilm. Inform.*, vol. 7, no. 01, p. 48, 2019, doi: 10.33884/jif.v7i01.1003.

[11] G. Gunawan, "Aplikasi Kamus Istilah Ekonomi (Inggris-Indonesia) Menggunakan Metode Sequential Searching," *Pseudocode*, vol. 3, no. 2, pp. 122–128, 2017, doi: 10.33369/pseudocode.3.2.122-128.

[12] A. Sonita and M. Sari, "Implementasi Algoritma Sequential Searching Untuk Pencarian Nomor Surat Pada Sistem Arsip Elektronik," *Pseudocode*, vol. 5, no. 1, pp. 1–9, 2018, doi: 10.33369/pseudocode.5.1.1-9.

[13] S. Tini, "Implementation of Sequential Search Method on Android-based Jakabaring Dictionary," *J. Transform.*, vol. 16, no. 1, p. 74, 2018, doi: 10.26623/transformatika.v16i1.830.

[14] M. Utami and Y. Apridiansyah, "Implementasi Algoritma Sequential Searching Pada Sistem Pelayanan Puskesmas Menggunakan Bootstrap (Studi Kasus Puskesmas Kampung Bali Bengkulu)," *JSAI (Journal Sci. Appl. Informatics)*, vol. 2, no. 1, pp. 81–86, 2019, doi: 10.36085/jsai.v2i1.166.

[15] I. Pratama, "Virtual Parallel Environment Using Pvm Case Study Bubble Sort Algorithm," *Proxies J. Inform.*, vol. 1, no. 2, pp. 44–53, 2017, [Online]. Available: http://103.243.177.137/index.php/proxies/article/view/1248.

[16] M. E. Al Rivan, "Perbandingan Kecepatan Gabungan Algoritma Quick Sort dan Merge Sort dengan Insertion Sort, Bubble Sort dan Selection Sort," *J. Tek. Inform. dan Sist. Inf.*, vol. 3, no. 2, pp. 319–331, 2017, doi: 10.28932/jutisi.v3i2.629.

[17] A. R. Lipu, R. Amin, M. N. I. Mondal, and M. Al Mamun, "Exploiting parallelism for faster implementation of Bubble sort algorithm using

FPGA," *ICECTE 2016 - 2nd Int. Conf. Electr. Comput. Telecommun. Eng.*, no. December, pp. 8–10, 2017, doi: 10.1109/ICECTE.2016.7879576.

[18] S. Roma Rio, M. Yusman, and F. Febi Eka, "Implementasi Algoritma Bubble Sort Dan Selection Sort Menggunakan Arraylist Multidimensi Pada Pengurutan Data Multi Prioritas," *J. Komputasi*, vol. 5, no. 1, pp. 81–87, 2017.