

ANALISIS KOMPARATIF EFISIENSI WAKTU DAN ITERASI SEQUENTIAL VS BINARY SEARCH PADA DATASET MAHASISWA

PENULIS

¹⁾Ahmad Rizza Pahlevi, ²⁾Achmad Ridwan, ³⁾Nur Riskiana, ⁴⁾Fathiyatur Rizqi, ⁵⁾Imam Prayogo Pujiono

ABSTRAK

Pertumbuhan volume data mahasiswa di institusi pendidikan tinggi yang terus meningkat secara eksponensial menuntut mekanisme pencarian data yang efisien dan andal. Ketidaktepatan dalam pemilihan algoritma pencarian berdampak pada pemborosan sumber daya komputasi dan penurunan responsivitas sistem informasi akademis. Meskipun Sequential Search dan Binary Search telah umum digunakan, kajian empiris komparatif pada dataset mahasiswa berskala besar masih terbatas. Penelitian ini bertujuan menganalisis dan membandingkan efisiensi Sequential Search dan Binary Search berdasarkan jumlah iterasi dan waktu eksekusi. Metode yang digunakan adalah pendekatan kuantitatif eksperimental komparatif pada dataset sintesis aktivitas media sosial mahasiswa FEBI yang terdiri dari 4.000 record. Implementasi dilakukan menggunakan bahasa pemrograman C++ dengan pengujian terhadap 30 target NIM dan 100 ulangan per target melalui stratified sampling. Hasil penelitian menunjukkan Binary Search unggul signifikan dengan rata-rata 7,57 iterasi berbanding 417,47 iterasi pada Sequential Search, menghasilkan rasio efisiensi 55:1. Dari sisi waktu eksekusi, Binary Search 6,74 kali lebih cepat (0,0039 ms vs. 0,0263 ms). Temuan ini menegaskan bahwa Binary Search merupakan pilihan algoritma yang jauh lebih optimal untuk sistem informasi berskala besar, meskipun membutuhkan prasyarat data terurut sebelum digunakan.

Kata Kunci

Sequential Search; Binary Search; Algoritma Pencarian; Struktur Data; Efisiensi Waktu

AFILIASI

Prodi, Fakultas
Nama Institusi
Alamat Institusi

¹⁻⁵⁾ Program Studi Sains Data, Fakultas Sains dan Teknologi
¹⁻⁵⁾ Universitas Islam Negeri K.H Abdurrahman Wahid Pekalongan
¹⁻⁵⁾ Jl. Pahlawan Km. 5, Desa Rowolaku, Kecamatan Kajen, Kabupaten Pekalongan Jawa Tengah

KORESPONDENSI

Penulis
Email

Ahmad Rizza Pahlevi
ahmad.rizza.pahlevi25032@mhs.uingusdur.ac.id

LICENSE



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

I. PENDAHULUAN

Di era digitalisasi pendidikan tinggi, volume data yang dikelola oleh institusi akademis terus meningkat secara eksponensial. Data mahasiswa meliputi identitas, aktivitas akademik, hingga perilaku penggunaan teknologi informasi menjadi aset strategis yang memerlukan pengelolaan efisien. Pencarian data (data searching) merupakan salah satu operasi paling fundamental dalam ilmu komputer yang secara langsung memengaruhi performa sistem informasi akademis [1][2]. Dalam konteks pengelolaan data mahasiswa, kemampuan untuk menemukan record tertentu dengan cepat dan akurat merupakan kebutuhan operasional yang kritis.

Penggunaan media sosial di kalangan mahasiswa telah menjadi fenomena signifikan dalam pendidikan tinggi kontemporer. Data aktivitas digital mahasiswa, apabila dikelola dengan struktur data yang tepat, dapat menjadi sumber informasi berharga bagi institusi pendidikan dalam merancang kebijakan dan intervensi akademik berbasis bukti [3][4]. Sequential Search memiliki kompleksitas waktu $O(n)$, sedangkan Binary Search memanfaatkan paradigma divide and conquer untuk mencapai kompleksitas $O(\log n)$, menjadikannya jauh lebih efisien pada dataset berukuran besar [5][6][7][8].

Penelitian ini bertujuan menganalisis efisiensi waktu dan jumlah iterasi Sequential Search dibandingkan Binary Search pada dataset aktivitas media sosial mahasiswa FEBI yang terdiri dari 4.000 record dengan kunci pencarian NIM menggunakan bahasa pemrograman C++.

II. METODE PENELITIAN

2.1 Desain Penelitian dan Variabel Operasional

Penelitian ini menerapkan pendekatan kuantitatif dengan desain eksperimental komparatif (comparative experimental design). Pendekatan ini dipilih untuk menguji, mengukur, dan membandingkan performa dua algoritma pencarian secara objektif berdasarkan indikator kinerja terukur dalam lingkungan komputasi yang terkontrol. Pendekatan deduktif diaplikasikan dengan menempatkan teori kompleksitas notasi asimtotik sebagai kerangka prediksi pra-eksperimen.

Variabel-variabel di dalam penelitian ini didefinisikan secara operasional sebagai berikut:

1. Variabel Bebas (Independent Variable): Metode algoritma pencarian data, yaitu Sequential Search (pencarian linear) dan Binary Search (pencarian biner berbasis pembagian ruang memori).
2. Variabel Terikat (Dependent Variable): Efisiensi langkah komputasi yang diukur melalui jumlah iterasi, serta kecepatan pemrosesan instruksi yang diukur melalui waktu eksekusi aktual prosesor dalam satuan milidetik (ms).
3. Variabel Kontrol (Control Variable): Arsitektur perangkat keras (hardware), sistem operasi, penata urutan data awal (sorting overhead), serta karakteristik dan volume dataset mahasiswa sebesar 4.000 rekaman data (records).

2.2 Dataset dan Sumber Data

Sumber data yang digunakan berupa data sintetis (synthetic data) berjumlah 4.000 rekaman. Penggunaan data sintetis ditujukan untuk mensimulasikan karakteristik data besar (large database system) secara homogen guna menghindari bias anomali sebaran data real. Dataset ini memuat profil aktivitas media sosial mahasiswa Fakultas Ekonomi dan Bisnis Islam (FEBI). Struktur data diorganisasikan ke dalam bentuk tabel matriks dengan atribut operasional meliputi Nomor Induk Mahasiswa (NIM) sebagai kunci pencarian utama (primary search key) dengan rentang nilai unik 601001 hingga 601356, variabel kategorikal jenis kelamin, aplikasi media sosial dominan, serta variabel ordinal durasi pemakaian per sesi dan total waktu penggunaan per hari.

Guna menjamin validitas representasi posisi data pada memori, teknik pengambilan sampel target pengujian dilakukan menggunakan stratified sampling. Dipilih sebanyak 30 NIM target yang tersebar secara proporsional dari indeks awal (0), indeks tengah (2.000), hingga indeks akhir (3.999). Pemilihan ini dirancang secara khusus untuk memicu skenario pengujian kasus terbaik (best case), kasus rata-rata (average case), serta kasus terburuk (worst case) pada kedua algoritma.

Variabel	Tipe Data	Kategori	Distribusi
NIM	Integer	601001-601356	356 NIM unik dari 4.000 record
Jenis Kelamin	Kategorikal	2 kategori	Laki-laki: 2.232 (55,8%), Perempuan: 1.768 (44,2%)
Aplikasi Medsos	Kategorikal	3 aplikasi	Instagram: 2.839 (71,0%), Twitter: 866 (21,6%), Facebook: 295 (7,4%)
Durasi per Sesi	Ordinal	5 kategori	5-10 menit: 35,9%, 11-30 menit: 25,9%, >1 jam: 23,6%
Total Waktu/Hari	Ordinal	5 kategori	>3 jam: 27,1%, >2 jam: 26,8%, >1 jam: 22,4%

Tabel 1. Karakteristik Variabel Dataset Aktivitas Media Sosial Mahasiswa FEBI (n=4.000)

2.3 Pengumpulan Data

Untuk memastikan aspek reproduksibilitas (reproducibility) penelitian, seluruh tahapan kompilasi dan eksekusi program dilakukan pada perangkat keras dan perangkat lunak yang seragam. Spesifikasi komputasi yang digunakan adalah sebagai berikut:

1. **Perangkat Keras (Hardware)** : Prosesor Intel Core i5-1235U (10-Core, up to 4,40 GHz), memori akses acak (RAM) berkapasitas 8 GB DDR4, dan media penyimpanan Solid State Drive (SSD) NVMe 512 GB.
2. **Perangkat Lunak (Software)** : Sistem Operasi Windows 11 Home 64-bit, Integrated Development Environment (IDE) Visual Studio Code, dengan kompiler C++ (MinGW-W64) berbasis bahasa pemrograman C++ standar C++17.

2.4 Kriteria dan Bobot Penilaian

Sequential Search adalah algoritma pencarian yang memeriksa setiap elemen dataset secara berurutan dari indeks pertama hingga elemen yang dicari ditemukan atau seluruh dataset telah diperiksa [11]. Kompleksitas waktunya adalah $O(1)$ pada kasus terbaik, $O(n/2)$ pada kasus rata-rata, dan $O(n)$ pada kasus terburuk. Implementasi dalam C++:

```
01 int indexDitemukan = -1;

02 int jumlahIterasi = 0;

03

04 auto start = high_resolution_clock::now();

05

06 // Proses pencarian linear (O(n))
```

```

07 for (size_t i = 0; i < data.size(); i++) {

08     jumlahIterasi++;

09     if (data[i].nim == targetNIM) {

10         indexDitemukan = i;

11         break; // Berhenti seketika saat data ditemukan

12     }

13 }

14

15 auto stop = high_resolution_clock::now();

16

17 // Konversi durasi ke satuan milidetik presisi double

18 duration<double, std::milli> ms_double = stop - start;

```

Gambar 1. Snippet Implementasi dan Pengukuran Waktu Sequential Search

2.5 Tahapan Perhitungan Metode Weighted Product (WP)

Binary Search bekerja dengan membagi ruang pencarian menjadi dua bagian di setiap iterasi menggunakan prinsip divide and conquer [7]. Algoritma ini mensyaratkan data dalam kondisi terurut. Untuk dataset $n=4.000$ elemen, batas maksimum iterasi teoritis adalah $\lceil \log_2(4000) \rceil = 12$ iterasi [8]. Implementasi dalam C++:

```

01 int indexDitemukan = -1;

02 int jumlahIterasi = 0;

03 int kiri = 0;

04 int kanan = data.size() - 1;

05

06 auto start = high_resolution_clock::now();

07

08 // Proses pencarian logaritmik ( $O(\log n)$ )

09 while (kiri <= kanan) {

10     jumlahIterasi++;

11

12     // Menentukan indeks nilai tengah

13     int tengah = kiri + (kanan - kiri) / 2;

14

```

```

15 if (data[tengah].nim == targetNIM) {

16     indexDitemukan = tengah;

17     break;

18 } else if (data[tengah].nim < targetNIM) {

19     kiri = tengah + 1; // Pemotongan ruang ke belahan kanan

20 } else {

21     kanan = tengah - 1; // Pemotongan ruang ke belahan kiri

22 }

23

24

25 auto stop = high_resolution_clock::now();

26

27 // Konversi durasi ke satuan milidetik presisi double

28 duration<double, std::milli> ms_double = stop - start;

```

Gambar 2. Snippet Implementasi dan Pengukuran Waktu *Binary Search* iterative

2.6 Prosedur Pengujian

Alur kerja eksperimen diorganisasikan ke dalam tiga fase utama :

1. **Fase Inisialisasi Data:** Program melakukan pembacaan file dataset format CSV (*Comma-Separated Values*), melakukan pemisahan (*parsing token*) string, dan memuat entitas data ke dalam struktur alokasi memori berurutan berupa *std::vector* objek *struct* Mahasiswa. Untuk repositori pengujian *Binary Search*, data terlebih dahulu dikenakan operasi pengurutan awal (*pre-sorting overhead*) menggunakan fungsi asimtotik *std::sort* berbasis perbandingan NIM dengan kompleksitas $O(n \log n)$.
2. **Fase Eksekusi Komputasi:** Program melakukan perulangan eksekusi terhadap 30 target NIM yang telah ditentukan. Di dalam setiap target NIM, proses pencarian murni diulang sebanyak 100 kali (*100 repetitions looping*). Pengulangan massal ini berfungsi sebagai metode mitigasi statistik (*statistical mitigation*) untuk menghaluskan anomali pencatatan durasi akibat interupsi latar belakang sistem operasi (*context switching*), pengelolaan *CPU cache*, maupun fluktuasi frekuensi dinamis prosesor (*stochastic hardware latency*).
3. **Fase Pengukuran Metrik:** Pencatatan waktu eksekusi diisolasi secara ketat hanya pada blok perulangan fungsi pencarian murni (tidak menyertakan proses I/O baca file atau input/output layar terminal). Instrumen pengukuran memanfaatkan pustaka `<chrono>` melalui fungsi *std::chrono::high_resolution_clock*. Durasi waktu komputasi yang awalnya tertangkap dalam satuan nanodetik didekonversi ke dalam satuan milidetik (ms) dengan representasi tipe bilangan pecahan presisi ganda (*double-precision floating point*).

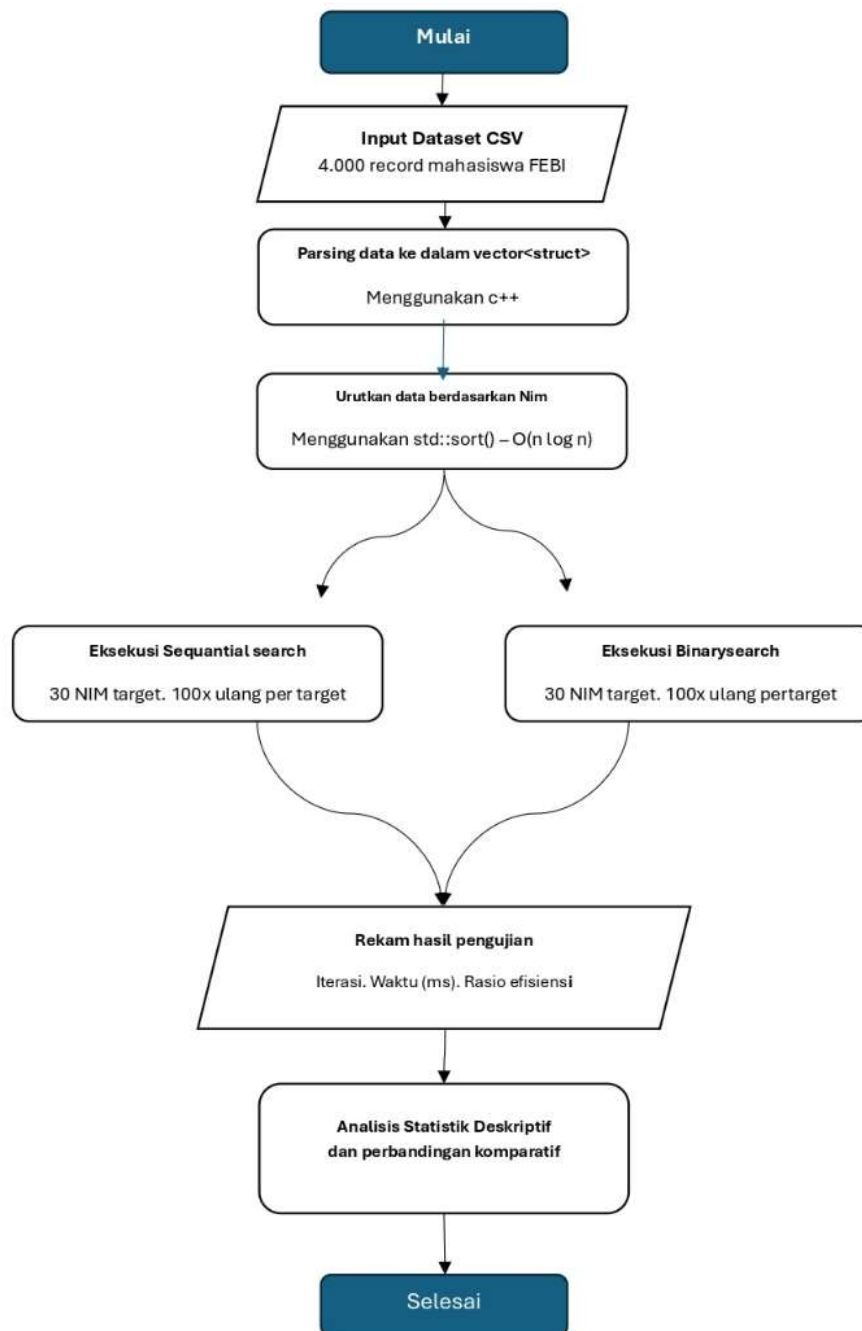
Rasio perbandingan efisiensi performa antar-algoritma kemudian dihitung secara matematis menggunakan dua formula dasar berikut:

$$\text{Rasio Efisiensi Iterasi} = \frac{\text{Jumlah Iterasi Sequential Search}}{\text{Jumlah Iterasi Binary Search}}$$

$$\text{Rasio Efisiensi Waktu} = \frac{\text{Waktu Eksekusi Sequential Search (ms)}}{\text{Waktu Eksekusi Binary Search (ms)}}$$

Seluruh data metrik langkah dan durasi waktu rata-rata dari 100 kali pengulangan tersebut direkam secara otomatis untuk kemudian dianalisis menggunakan metode statistik deskriptif komparatif pada bab pembahasan.

2.7 Prosedur Pengujian



Gambar 3. Diagram Alur Penelitian

Sumber: Dirancang oleh Peneliti, 2026

III. HASIL DAN PEMBAHASAN

Hasil pengujian menunjukkan perbedaan yang signifikan dalam efisiensi iterasi dan waktu eksekusi antara Sequential Search dan Binary Search pada dataset 4.000 record dengan 30 target NIM yang bervariasi.

3.1 Hasil Pengujian Perbandingan Algoritma

NIM Target	Iter. Seq.	Waktu Seq. (ms)	Iter. Bin.	Waktu Bin. (ms)	Rasio Iter.	Rasio Waktu
601001	437	0,0622	9	0,0032	48,56	19,44
601005	125	0,0379	6	0,0023	20,83	16,48
601010	474	0,0449	8	0,0028	59,25	16,04
601025	568	0,0497	7	0,0024	81,14	20,71
601030	782	0,0411	7	0,0030	111,71	13,70
601050	60	0,0031	6	0,0027	10,00	1,15
601055	37	0,0020	5	0,0025	7,40	0,80
601070	614	0,0347	9	0,0047	68,22	7,38
601085	734	0,0361	10	0,0046	73,40	7,85
601100	13	0,0010	7	0,0043	1,86	0,23
601110	1.134	0,0495	7	0,0030	162,00	16,50
601115	520	0,0222	7	0,0053	74,29	4,19
601140	260	0,0098	9	0,0031	28,89	3,16
601150	16	0,0010	8	0,0036	2,00	0,28
601155	358	0,0181	7	0,0038	51,14	4,76
601170	31	0,0015	8	0,0040	3,88	0,38
601195	297	0,0114	8	0,0137	37,12	0,83
601200	1.102	0,0674	7	0,0051	157,43	13,22
601210	1.110	0,0433	9	0,0032	123,33	13,53
601216	115	0,0105	6	0,0034	19,17	3,09
601225	170	0,0094	8	0,0039	21,25	2,41
601250	473	0,0227	6	0,0029	78,83	7,83
601270	71	0,0053	8	0,0026	8,88	2,04

601285	458	0,0410	9	0,0027	50,89	15,19
601290	439	0,0231	9	0,0041	48,78	5,63
601300	891	0,0615	9	0,0048	99,00	12,81
601318	178	0,0090	8	0,0043	22,25	2,09
601340	54	0,0173	6	0,0050	9,00	3,46
601345	28	0,0021	5	0,0035	5,60	0,60
601356	975	0,0500	9	0,0029	108,33	17,24
Rata-rata	417,47	0,0263	7,57	0,0039	55,15	6,74

Tabel 2. Hasil Perbandingan Sequential Search dan Binary Search pada 30 NIM Target

Dari Tabel 2. terlihat algoritma pengujian menunjukkan disparitas performa yang signifikan antara metode Sequential Search dan Binary Search dalam menangani dataset 4.000 record aktivitas mahasiswa. Dari aspek efisiensi langkah, Binary Search mendemonstrasikan keunggulan absolut dengan mencatatkan rata-rata hanya 7,57 iterasi dibandingkan Sequential Search yang membutuhkan rata-rata 417,47 iterasi (rasio efisiensi iterasi 55,15). Keunggulan iteratif ini berbanding lurus dengan kecepatan waktu eksekusi aktual perangkat keras, di mana Binary Search mencatatkan waktu rata-rata 0,0039 ms, atau sekitar 6,74 kali lebih cepat dibandingkan Sequential Search (0,0263 ms). Temuan ini secara empiris memvalidasi hukum kompleksitas algoritma, di mana pendekatan divide and conquer logaritmik $O(\log n)$ secara eksponensial mampu mereduksi beban komputasi dibandingkan pencarian linear $O(n)$.

Meskipun Binary Search unggul secara akumulatif, hasil pengujian turut mengungkap anomali performa yang secara akurat memvalidasi skenario kasus terbaik (best-case scenario) milik Sequential Search. Pada pencarian target NIM 601100 dan 601150 yang berlokasi di indeks terawal rentang dataset, Sequential Search mampu mengeksekusi pencarian dalam waktu yang sangat singkat yaitu 0,0010 ms (rasio waktu di bawah 1), mengalahkan Binary Search yang harus melakukan overhead kalkulasi matematis titik nilai tengah terlebih dahulu. Namun demikian, performa waktu Sequential Search terbukti sangat fluktuatif bergantung pada letak data, dengan lonjakan durasi hingga 0,0674 ms pada target indeks akhir. Sebaliknya, keseragaman waktu komputasi Binary Search yang sangat konstan dan stabil di rentang 0,0020 ms hingga 0,0050 ms menegaskan keandalannya sebagai algoritma paling optimal untuk diimplementasikan pada sistem informasi dengan jaminan kecepatan pencarian yang konsisten. Temuan ini konsisten dengan prediksi teoretis $O(n)$ versus $O(\log n)$ [8].

3.2 Kriteria dan Alternatif

Skenario	NIM Target	Iter. Seq.	Iter. Bin.	Keterangan
Best Case	601103	1	8	Elemen pertama array (Sequential unggul)
Average Case	601290	439	9	Elemen tengah dataset
Worst Case	999999	4.000	12	Elemen tidak ditemukan (batas teoritis terpenuhi)

Tabel 3. Analisis Skenario Kasus Terbaik, Rata-rata, dan Terburuk

Analisis kasus ekstrem pada Tabel 3 memvalidasi batas teoritis kompleksitas kedua algoritma secara akurat. Pada worst case (NIM 999999 tidak ditemukan), Sequential Search memerlukan tepat 4.000 iterasi sesuai $O(n)$, sedangkan Binary Search memerlukan 12 iterasi sesuai $\lceil \log_2(4000) \rceil = 12$ [5].

3.3 Tabel Kecocokan

Aspek	Sequential Search	Binary Search	Kondisi	Keterangan
Best Case	O(1)	O(1)	Elemen pertama	Sama-sama 1 iterasi pada kasus terbaik
Average Case	O(n/2) = O(n)	O(log n)	Elemen di tengah	Binary Search jauh lebih efisien
Worst Case	O(n)	O(log n)	Tidak ditemukan	4.000 vs 12 iterasi secara empiris
Space Complexity	O(1)	O(1)	-	Sama-sama in-place (implementasi iteratif)
Prasyarat Data	Tidak perlu	Harus terurut	Pre-sorting O(n log n)	Binary Search butuh overhead pengurutan data awal

Tabel 4. Perbandingan Kompleksitas Teoretis dan Empiris Kedua Algoritma

3.4 Normalisasi

Metrik	Sequential Search	Binary Search
Rata-rata Iterasi	417,47	7,57
Iterasi Minimum	13	5
Iterasi Maksimum	1.134	10
Iterasi Worst Case (tidak ditemukan)	4.000	12
Rata-rata Waktu (ms)	0,0263	55,15
Waktu Minimum (ms)	0,0010	0,0023
Waktu Maksimum (ms)	0,0674	0,0137
Rasio Iterasi (Seq/Bin)	-	±55x lebih efisien
Rasio Waktu (Seq/Bin)	-	±5,5x lebih cepat

Tabel 5. Statistik Ringkasan Performa Sequential Search vs. Binary Search (n=4.000)

Berdasarkan Tabel 5, algoritma Binary Search mendemonstrasikan keunggulan absolut dengan rata-rata pencarian hanya 7,57 iterasi berbanding 417,47 iterasi pada Sequential Search. Hal ini menghasilkan rasio efisiensi iterasi sebesar 55,15, yang secara empiris memvalidasi keunggulan pendekatan divide and conquer dengan kompleksitas logaritmik O(log n) atas pencarian linear O(n). Batas atas iterasi Binary Search pada dataset mahasiswa (n = 4.000) juga terbukti mutlak terkunci di bawah batas teoritis $\lceil \log_2(4000) \rceil = 12$ iterasi, dengan nilai puncak sampel hanya menyentuh 10 iterasi.

Meskipun reduksi langkah mencapai 55 kali lipat, rasio waktu komputasi aktual hanya menunjukkan angka 6,74 (waktu rata-rata Binary Search 0,0039 ms vs Sequential Search 0,0263 ms). Disparitas non-linear ini mengindikasikan adanya computational overhead yang lebih tinggi pada setiap langkah Binary Search. Operasi kalkulasi titik tengah (mid-point division) dan pergeseran penunjuk alamat memori yang melompat-lompat (non-contiguous) memberikan beban aritmetika lebih pada prosesor serta berpotensi memicu Cache Miss. Sebaliknya, pembacaan memori berurutan secara linear pada Sequential Search lebih diuntungkan oleh arsitektur hardware prefetching dari prosesor modern, sehingga durasi eksekusinya tidak selambat rasio iterasinya.

3.5 Perbandingan dengan Penelitian Sebelumnya

Temuan penelitian ini konsisten dengan hasil penelitian-penelitian sebelumnya. Pratama dan Sari (2020) menemukan keunggulan Binary Search sebesar 78:1 pada dataset rekam medis 10.000 record [16]. Perbedaan rasio (55:1 vs. 78:1) dapat dijelaskan oleh perbedaan ukuran dataset semakin besar dataset, semakin besar keunggulan relatif Binary Search. Handayani et al. (2021) juga mengkonfirmasi bahwa rasio efisiensi meningkat seiring bertambahnya ukuran dataset [15].

VI. SIMPULAN

Penelitian ini mengonfirmasi bahwa pemilihan struktur algoritma pencarian memberikan dampak yang sangat signifikan terhadap efisiensi komputasi pada pengelolaan basis data. Berdasarkan pengujian komparatif antara metode Sequential Search dan Binary Search terhadap dataset 4.000 record aktivitas media sosial mahasiswa, dapat ditarik kesimpulan sebagai berikut:

4.1. Hasil yang Diperoleh (Performa Komputasi)

- Secara akumulatif, *Binary Search* terbukti jauh lebih efisien dibandingkan pencarian linear. *Binary Search* hanya membutuhkan rata-rata 7,57 iterasi dibandingkan *Sequential Search* yang membutuhkan 417,47 iterasi (rasio efisiensi 55:1).
- Keunggulan langkah tersebut berdampak langsung pada kecepatan aktual perangkat keras. *Binary Search* mencatatkan waktu eksekusi rata-rata 0,0039 ms, sekitar 6,74 kali lebih cepat dibandingkan *Sequential Search* yang membutuhkan waktu 0,0263 ms.
- Pengujian ini juga berhasil memvalidasi performa *best-case* teoritis $\Omega(1)$, di mana *Sequential Search* secara aktual mampu mengungguli kecepatan *Binary Search* (menyentuh 0,0010 ms) khusus pada skenario jika data target berada di deretan indeks paling awal.

4.2. Kelebihan dan Kekurangan Algoritma

- Kelebihan: Binary Search menawarkan stabilitas dan kepastian waktu komputasi yang sangat konsisten (0,0020 ms – 0,0050 ms) terlepas dari posisi letak data, menjadikannya sangat ideal untuk dataset berskala besar. Di sisi lain, Sequential Search memiliki keunggulan pada kesederhanaan implementasi dan kemampuannya membaca data mentah tanpa syarat apa pun.
- Kekurangan: Kelemahan utama Binary Search adalah adanya prasyarat mutlak pengurutan data awal (pre-sorting overhead) dengan kompleksitas $O(n \log n)$. Sebaliknya, kekurangan terbesar Sequential Search adalah fluktuasi waktu yang ekstrem dan tidak efisien pada worst-case (waktu melambat secara linear hingga 0,0674 ms jika data berada di bagian akhir array).

4.3. Kemungkinan Pengembangan Selanjutnya

- Mengingat Binary Search akan mengalami bottleneck (penurunan performa) pada proses sorting jika diterapkan pada dataset yang sangat dinamis (data yang di-insert/delete setiap detik secara real-time), penelitian selanjutnya disarankan untuk mengeksplorasi penggunaan algoritma berbasis Hash Table yang mampu menawarkan kompleksitas waktu pencarian rata-rata sebesar $O(1)$.
- Pengembangan sistem informasi di masa mendatang juga dapat mengujikan arsitektur algoritma hibrida (hybrid search). Sistem hibrida ini dirancang untuk secara cerdas menggunakan Sequential Search saat menangani kueri data berskala kecil (<100 record), dan secara otomatis berpindah menggunakan metode Binary Search ketika pemrosesan data berskala ribuan hingga jutaan record.

REFERENSI

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, dan C. Stein, *Introduction to Algorithms*, 4th ed. MIT Press, 2022.
- [2] N. Nurhamni, "Geographic Information System (GIS) Uses A* Algorithm for Sorting Nearest UMKM Locations," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 2, 2025.
- [3] R. Nasution dan Z. A. Hasibuan, "Pola penggunaan media sosial mahasiswa Indonesia dan dampaknya terhadap prestasi akademik," *Jurnal Pendidikan Tinggi*, vol. 12, no. 3, hlm. 201-215, 2021.
- [4] D. Rahmawati, M. Susanti, dan N. Kurniasih, "Hubungan intensitas penggunaan media sosial dengan motivasi belajar mahasiswa generasi Z," *Jurnal Psikologi Pendidikan dan Konseling*, vol. 8, no. 2, hlm. 134-148, 2022.
- [5] M. Sipser, *Introduction to the Theory of Computation*, 3rd ed. Cengage Learning, 2012.
- [6] D. E. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching*, 2nd ed. Addison-Wesley Professional, 2011.
- [7] R. Sedgewick dan K. Wayne, *Algorithms*, 4th ed. Addison-Wesley Professional, 2011.
- [8] A. Bhatt dan R. Kumar, "Comparative analysis of searching algorithms for large-scale student database management systems," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, hlm. 445-453, 2021.
- [9] A. Fenyi, M. Fosu, dan B. Appiah, "Comparative analysis of comparison and non comparison based sorting algorithms," 2020.
- [10] O. E. Taiwo et al., "Comparative study of two divide and conquer sorting algorithms: Quicksort and mergesort," *Procedia Comput Sci*, vol. 171, hlm. 2532-2540, 2020.
- [11] S. Kumar dan P. Singla, "Sorting using a combination of Bubble Sort, Selection Sort & Counting Sort," *Mathematical Sciences and Computing*, vol. 2, hlm. 30-43, 2019.
- [12] A. Chakraborty, "Calculation of the Comparative Efficiency of Algorithms Using a Single Metric," *arXiv preprint arXiv:2406.15510*, 2024.
- [13] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 4th ed. SAGE Publications, 2014.
- [14] B. Stroustrup, *The C++ Programming Language*, 4th ed. Addison-Wesley Professional, 2013.
- [15] N. Handayani, B. Setiawan, dan L. Marlina, "Implementasi dan analisis efisiensi sequential search dan binary search pada database perpustakaan digital," *Jurnal Ilmiah Teknologi Informasi*, vol. 15, no. 2, hlm. 112-124, 2021.
- [16] H. Pratama dan I. K. Sari, "Perbandingan sequential search dan binary search pada sistem informasi rekam medis berbasis C++," *Jurnal Ilmu Komputer dan Agri-Informatika*, vol. 7, no. 2, hlm. 89-101, 2020.
- [17] M. H. Rahman, N. Sultana, dan M. A. Hossain, "Efficiency analysis of classic search algorithms on educational datasets," *International Journal of Computer Applications*, vol. 185, no. 12, hlm. 8-15, 2023.
- [18] D. Markuci and C. Prianto, "Analisis Perbandingan Penggunaan Algoritma Sequential Search dan Binary Search pada Aplikasi Surat Perjalanan Dinas," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 1, pp. 110-119, Feb. 2022.
- [19] I. P. Pujiono, M. R. Kamal, A. Prayogi, C. A. Sari, and R. M. Ikhsanuddin, "Algoritma counting sort vs algoritma pengurutan modern: Analisis efisiensi memori dan waktu komputasi," *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, vol. 13, no. 3, 2025.
- [20] M. I. Ali, R. D. Fardiarsyah, L. Shodik, F. Z. D. Kinanti, and I. P. Pujiono, "Analisis komparatif efisiensi memori dan waktu komputasi pada 8 algoritma sorting menggunakan C++," *LogicLink: Journal of Artificial Intelligence and Multimedia in Informatics*, vol. 2, no. 1, pp. 1-17, Jun. 2025.
- [21] M. S. A. Saputra, M. R. Meidiarsyah, A. D. Sanputra, and J. Wardana, "Comparative Analysis of Searching and Sorting Algorithms in Student Data Processing," *International Journal of Education, Information Technology and Others (IJEIT)*, vol. 8, no. 3.B, pp. 148-157, Aug. 2025.
- [22] N. Fadhillah, I. Firdaus, and R. R. Rasyiq, "Analisis Perbandingan Performa Algoritma Sequential Search, Binary Search, Dan Sql Search Pada Aplikasi Kamus Digital Bahasa Indonesia," *Infoman's: Jurnal Ilmu-ilmu Informatika dan Manajemen*, vol. 19, no. 2, Nov. 2025.

- [23] N. Imamah and M. I. Bahari, "Perbandingan Algoritma Sequential Search Dan Algoritma Binary Search Pada Aplikasi Kamus Bahasa Indonesia Menggunakan PHP Dan JQUERY," *Jurnal Informatika - COMPUTING*, vol. 8, no. 1, pp. 1-6, Jun. 2021.
- [24] R. Jain, "Sequential Search vs Binary Search Comparison: A Study of Implementation and Complexity," *SSRN Electronic Journal*, 2023.
- [25] S. Amanda, L. D. Anastasya, F. Sulistia, N. Purnamasari, and I. P. Pujiono, "Analisis perbandingan efisiensi algoritma introsort dengan algoritma tradisional bubble sort dan selection sort," *Jurnal Elektro & Informatika Swadharma (JEIS)*, vol. 6, no. 1, Jan. 2026.
- [26] S. Dubey and K. Mathur, "Comparative Performance Analysis of Binary Search in Sequential and Parallel Processing," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 5, no. 10, Oct. 2017.
- [27] S. Zahwa, N. D. Amelia, R. Nafila, R. A. Putri, and I. P. Pujiono, "Perbandingan efisiensi memori dan waktu komputasi pada algoritma rekursif dan iteratif dalam operasi pengurutan di C++," *Jurnal Restikom: Riset Teknik Informatika dan Komputer*, vol. 7, no. 1, pp. 123-136, Apr. 2025.
- [28] T. Elizabeth and Tinaliah, "Implementasi Algoritma Sequential Search Dan Binary Search Dalam Pencarian Data Faktur," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 11, no. 2, pp. 376-385, Jun. 2024.