JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.5948

COMPARISON OF THE COMPLEXITY OF SEARCH ALGORITHMS IN DIGITAL PAPUAN LANGUAGE DICTIONARIES

Nur Fitrianingsih Hasan[1*]; Vera Wati[2]; Nurfadillah[1]; Bima Julianto Mambobo[1]

Computer Science[1]
Universitas Muhammadiyah Papua, Jayapura City, Indonesia[1]
https://umpapua.ac.id[1]
hi.fitri@umpapua.ac.id[*]

Smart City Information Systems[2]
Universitas Tunas Pembangunan, Surakarta, Indonesia[2]
https://utp.ac.id[2]
verave.wati@gmail.com

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract—*Regional-language digital dictionaries play a strategic role in supporting Tourism 4.0 by enabling communication between local communities and tourists. However, inefficient search mechanisms can substantially degrade their usability, while research on the computational complexity of search algorithms for low-resource languages such as those of Papua remains scarce. This study presents a comparative analysis of three search algorithms binary search, tree-based search, and n-gram search benchmarked against an unoptimized linear search baseline in a Papuan regional-language digital dictionary containing 5,597 lemmas. Each algorithm was evaluated on both time and space complexity through controlled experiments executed on five heterogeneous computing devices. The experimental results show that the tree-based search algorithm achieves the best overall performance, with the lowest average search time of 1.28 seconds and the smallest average memory usage of 3.73 kB. These findings provide an empirical basis for selecting efficient search algorithms in regional-language digital dictionaries and contribute to the Tourism 4.0 digital infrastructure by enabling fast, scalable access to local-language information.*

Keywords: *Algorithm Complexity, Digital Dictionary, Low-Resource Languages, Tourism 4.0, Tree-Based Search.*

Intisari— *Kamus digital bahasa daerah berperan strategis dalam mendukung Tourism 4.0 dengan memfasilitasi komunikasi antara masyarakat lokal dan wisatawan. Namun, mekanisme pencarian yang tidak efisien dapat menurunkan kegunaannya, sementara penelitian mengenai kompleksitas komputasi algoritma pencarian untuk bahasa dengan sumber daya terbatas (low-resource languages) seperti bahasa-bahasa Papua masih terbatas. Penelitian ini melakukan analisis komparatif tiga algoritma pencarian yaitu binary search, tree-based search, dan n-gram search dengan pembanding linear search tanpa optimasi pada kamus digital bahasa daerah Papua yang berisi 5.597 lema. Setiap algoritma dievaluasi berdasarkan kompleksitas waktu dan ruang melalui eksperimen terkontrol pada lima perangkat komputasi dengan spesifikasi heterogen. Hasil eksperimen menunjukkan bahwa algoritma tree-based search memberikan kinerja terbaik dengan waktu pencarian rata-rata tercepat 1,28 detik dan penggunaan memori rata-rata terkecil 3,73 kB. Temuan ini memberikan dasar empiris dalam pemilihan algoritma pencarian yang efisien untuk kamus digital bahasa daerah serta berkontribusi pada infrastruktur digital Tourism 4.0 melalui akses informasi bahasa lokal yang cepat dan skalabel.*

Kata Kunci: *Kompleksitas Algoritma, Kamus Digital, Bahasa Berdaya Rendah (Low-Resource Languages), Pariwisata 4.0, Pencarian Berbasis Pohon.*

INTRODUCTION

The first mission of the 2019–2023 Papua Province Tourism Office Strategic Plan is to strengthen tourism quality and competitiveness by internalizing Papua's traditional and cultural values, including the use of regional languages to support community empowerment in the tourism sector. This mission is increasingly urgent given that 19 Papuan regional languages have been reported as endangered [1]. The use of regional languages in tourism activities strengthens cultural identity and increases local community involvement in tourism development. However, practical barriers persist: not all tourists understand regional languages, regional vocabularies often lack terms for modern tourism objects, and exclusive use of regional languages can limit market reach [2]. Within Indonesia's linguistic landscape of over 700 languages, many regional languages remain underrepresented in computational resources, posing significant challenges for natural language processing and digital preservation [3], [4].

Tourism 4.0 integrates digital technologies, artificial intelligence, and data-driven systems to enhance tourist experiences and information accessibility [5], [6]. Recent systematic reviews show that Industry 4.0 paradigms are reshaping the tourism sector across multiple dimensions, including visitor-technology interaction, digital competencies, and smart-destination development [7], [8], [9]. Within this paradigm, mobile applications and intelligent information systems mediate communication between tourists and local communities [10]. For culturally rich regions such as Papua, digital language resources, particularly regional-language dictionaries, become enabling infrastructure that allows visitors to interact meaningfully with local communities while preserving linguistic heritage [11], [12].

The core operation of any digital dictionary is the search process, by which lexical entries are retrieved from a structured corpus. The efficiency of this operation is governed by the underlying search algorithm and is conventionally characterized along two axes: *time complexity*, which measures the growth of execution time as a function of input size *n*, and *space complexity*, which measures the auxiliary memory required during execution [13], [14]. In algorithm analysis, both are typically expressed using Big-O notation to describe worst-case bounds independent of hardware [14], [15]. Standard algorithms such as linear search ($O(n)$), binary search ($O(\log n)$ with sorted data), tree-based search ($O(\log n)$ with hierarchical indexing), and n-gram search (whose cost scales with substring token combinations) thus offer markedly different performance profiles when applied to dictionary retrieval [16], [17], [18].

The choice of search algorithm has direct, measurable consequences for Tourism 4.0 deployment. Tourism 4.0 services are typically delivered through mobile and web clients running on heterogeneous user devices, ranging from high-end smartphones to entry-level handsets with constrained CPU, memory, and network bandwidth [19]. On such platforms, user-perceived latency is a critical determinant of adoption: empirical evidence indicates that response times exceeding a few seconds substantially increase the likelihood of user abandonment, while delays as small as a few hundred milliseconds measurably reduce engagement [20]. An algorithm with poor time complexity, therefore, multiplies user-perceived latency on low-end devices, while an algorithm with poor space complexity restricts scalability as the lexical corpus grows or as the dictionary is integrated into resource-shared tourism applications [21]. Consequently, evaluating both time and space complexity, and doing so under heterogeneous hardware conditions, is not a peripheral engineering concern but a prerequisite for designing dictionary services that remain responsive across the device diversity characteristic of Tourism 4.0 secosystems.

Although digital dictionaries for regional languages have been studied extensively [11], [12], [22], [23], [24], [25], most prior work emphasizes system implementation or usability rather than computational performance. Where algorithms are examined, evaluations are usually confined to a single algorithm, such as binary search [26], brute force [27], or sequential search [24], or to time complexity alone, with limited reporting on space complexity, on testing protocols, or on cross-device behavior. A representative example is the Papuan regional-language digital dictionary developed by Hasan and Iribaram, which achieved 89% expert validation by the Balai Bahasa Provinsi Papua (BBPP) but exhibited slow data retrieval and lacked an evaluated search-algorithm layer [11]. To date, no published study has provided a comparative, hardware-aware complexity analysis of search algorithms specifically for low-resource Papuan-language digital dictionaries.

This study addresses that gap through a comparative analysis of *binary search*, *tree-based search*, and *n-gram search*, benchmarked against an unoptimized *linear search* baseline, using real dictionary data of 5,597 lemmas. The novelty of the work lies in three combined elements: (i) joint

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.5948

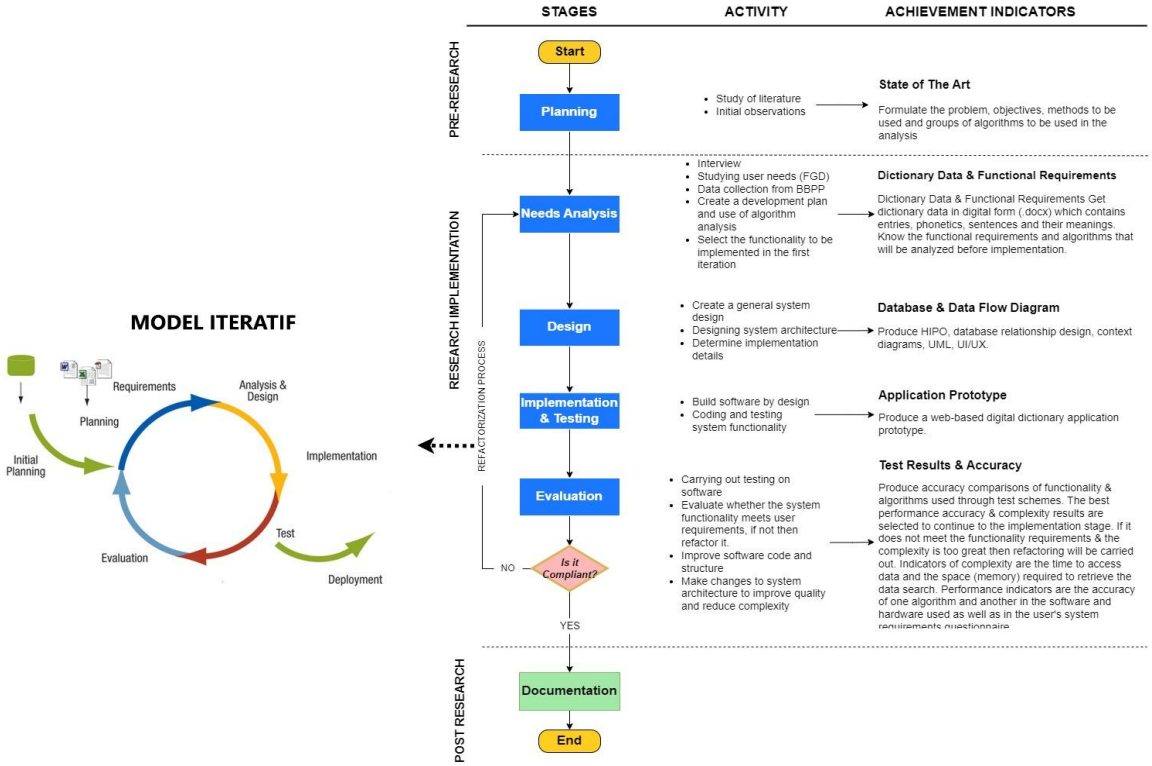
evaluation of *time and space complexity* rather than time alone; (ii) execution on *five heterogeneous computing devices* to reflect realistic Tourism 4.0 deployment conditions; and (iii) framing the algorithm-selection problem explicitly within the constraints of low-resource regional-language corpora. The empirical contribution is a hardware-aware recommendation for search-algorithm selection in regional-language digital dictionaries.

The remainder of this paper is organized as follows. Section II describes the materials and methods, including the dataset, the algorithms under test, the testing protocol, and the measurement tools. Section III presents the experimental results and a Big-O–grounded discussion. Section IV concludes the study and outlines directions for future work.

MATERIALS AND METHODS

This study adopts an iterative experimental approach in which the digital dictionary application, search-algorithm implementations, and testing protocol are refined across successive development cycles before final performance measurements are collected. Similar iterative approaches have been adopted in recent regional-language dictionary development projects in Indonesia [25]. The overall research workflow, illustrated in Figure 1, comprises three phases: pre-research (literature study and initial observation at Balai Bahasa Provinsi Papua), research implementation (needs analysis, design, implementation, testing, and evaluation), and post-research (documentation and dissemination).

The research stages, as depicted in Figure 1, are broadly divided into pre-research, research implementation, and post-research categories. Each stage and activity has achievement targets and outcomes. In the pre-research phase, literature reviews were conducted to produce the State of the Art (SoTA), and initial observations at Balai Bahasa Provinsi Papua were carried out to identify user needs and application requirements. The following steps in the implementation phase include needs analysis, design, implementation, testing, and evaluation (with refactoring as necessary). At this stage, the core of this research is the exploration of search algorithms for the Papuan regional-language dictionary. In the implementation of research activities, the core deliverables include system architecture requirements, algorithms, HIPO diagrams, data-flow design, databases, UI/UX, application prototypes, and application testing. The results of testing and evaluation of applications that meet the requirements serve as the basis for final implementation, after which the documentation phase produces the research outcomes.



Source : (Research Results, 2023)

Figure 1. Research Method

Dataset

The experimental dataset consists of 5,597 lemmas extracted from the Papuan regional-language digital dictionary previously developed and validated by Balai Bahasa Provinsi Papua (BBPP) [11]. Each lemma record contains a headword, definition, example sentence, and morphological variations. The dataset is stored in a MySQL database with alphabetical indexing on the headword column. The same dataset is used identically across all algorithms and all testing devices, ensuring that observed performance differences reflect algorithmic behavior rather than dataset variation. This dataset represents authentic low-resource language characteristics, in contrast to large-scale empirical evaluations that typically rely on big-data corpora containing millions of records [28].

Test Query Selection

Three representative query words were selected to reflect different positions in the alphabetically ordered dataset, following a positional sampling strategy: "Bahasa" (early position, letter B), "Makan" (middle position, letter M), and "Yospan" (late position, letter Y). This selection was designed to expose algorithm behavior across the full alphabetic range of the dataset, since several search algorithms exhibit position-dependent performance, for example, linear search performs more comparisons when the target lies near the end of the sequence, while binary and tree-based search depend on the path length from the search-tree root to the target node. Additionally, the three words differ in corpus frequency: "Bahasa" and "Makan" are high-frequency terms that appear in many example sentences, whereas "Yospan" is a culturally specific Papuan term with a single dedicated entry, enabling observation of how each algorithm handles both common and rare lexical entries.

Algorithm Implementation

All four search procedures were implemented in PHP within the Laravel framework, executing queries against the same MySQL table. The four procedures are:

1. *Baseline linear search.* Sequential scan of all lemma records without algorithmic optimization, retrieving any record whose headword or example sentence contains the query substring. Worst-case time complexity is *O(n)* [24], [16].
2. *Binary search.* Operates on the alphabetically sorted lemma collection through successive midpoint comparisons until the query is located or the search range is exhausted. Expected time complexity is *O(log n)* for sorted input [22], [26]
3. *Tree-based search.* Leverages hierarchical indexing on the headword column, with tree traversal delegated to the database engine. This enables logarithmic-time lookup, with expected complexity *O(log n)* [13], [15].
4. *N-gram search.* Performs substring matching by decomposing both query and stored headwords into character n-grams, enabling partial-match retrieval. Time complexity depends on the number of generated tokens [27].

All algorithms operate on the identical dataset under identical query conditions; differences in returned result sets (Section III) therefore reflect intrinsic algorithmic behavior rather than data or query variation.

Testing Protocol

Testing was conducted directly on the deployed web-based dictionary application. For each combination of ($algorithm \times query\ word \times testing\ devices$), the query was issued through the application's search interface, and the resulting performance metrics were recorded for analysis. This yields a total of 4 algorithms × 3 query words × 5 devices = 60 measured executions. The comparative-evaluation framework adopted here, measuring execution time and memory usage per algorithm, is consistent with recent empirical studies on classical search algorithms, including the large-scale evaluation by Purnama on linear, binary, and hash search using big-data corpora [28]. To minimize background interference, non-essential applications were closed during testing, and all devices were operated under stable network and power conditions.

Performance Measurement Tools

Two performance metrics were recorded for each query execution:

- Execution time (time complexity proxy). Captured at the application layer using PHP's `microtime(true)` function, with timestamps taken immediately before and after the search routine, then reported as the difference in seconds.
- Memory consumption (space complexity proxy). Captured using PHP's `memory_get_usage()` function, which records the amount of memory allocated by

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.5948

the PHP script. Snapshots are taken before and after the search routine, and the delta is reported in kilobytes (kB).

Both `microtime(true)` and `memory_get_usage()` are part of the PHP standard library and provide microsecond and byte-level resolution, respectively, which is adequate for the search workloads under study. The measured values were observed through the browser's developer tools (Network and Console panels) and recorded for analysis.

Testing Devices

Tests were conducted on five computing devices with heterogeneous hardware specifications, summarized in Table 1. The devices span different processor families (Intel Core i3/i5/i7 across generations 3, 7, and 11; Apple M1), RAM capacities from 4 GB to 16 GB, and storage tiers from 128 GB to 1 TB SSD. This heterogeneity reflects realistic Tourism 4.0 deployment conditions in which end-user devices vary substantially in computational capacity [19], [21].

Table 1. Testing Devices

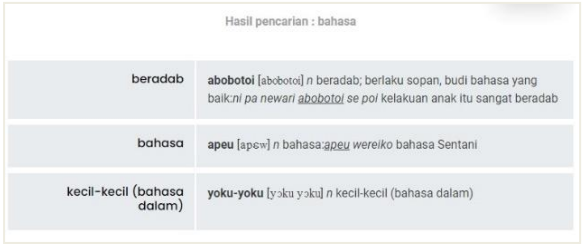
Device Name	Merk	Processor	RAM	Storage
Dev1	Assembled PC	Intel core i7 gen 3	16 GB	SSD 256GB
Dev2	ASUS	Intel core i3 gen 7	12 GB	SSD 128GB
Dev3	Lenovo	Intel core i7 gen 11	16 GB	SSD 1TB
Dev4	MacbookPro	M1	8 GB	SSD 256GB
Dev5	MacbookAir	Intel core i5	4 GB	SSD 256GB

Source : (Research Results, 2023)

RESULTS AND DISCUSSION

Search Accuracy Across Algorithms

Three test queries, "Bahasa," "Makan," and "Yospan," were issued to each algorithm to observe both correctness and matching behavior. Figures 2–5 illustrate the result sets returned for the query "Bahasa" under each algorithm.



Source : (Research Results, 2023)
Figure 2. In the Absence of an Algorithm

Figure 2 shows the search result for "Bahasa" using the dictionary without algorithmic optimization. The result set includes all records in which the substring "Bahasa" appears anywhere, including within example sentences, producing noisy output containing semantically unrelated entries.

Source : (Research Results, 2023)
Figure 3. Binary Algorithm

Figure 3 shows the result using the binary search algorithm. The algorithm returns records containing "Bahasa" both as headwords and within affixed forms, performing structured comparisons on the sorted dataset [26].

Source : (Research Results, 2023)
Figure 4. Tree Algorithm

Figure 4 shows the result using the tree-based search algorithm. Only a single record corresponding exactly to the headword "Bahasa" is returned, reflecting the hierarchical lookup behavior on the indexed headword column.

Source : (Research Results, 2023)
Figure 5. N-gram Algorithm

Figure 5 shows the result using the n-gram search algorithm. Two records are returned, corresponding to lemmas whose character n-grams

overlap with those of the query, demonstrating the algorithm's substring-matching nature. The patterns observed for "Bahasa" generalize across the three test queries. The query "Yospan" produces a single result across all algorithms because only one lemma corresponds to that headword, illustrating how dataset characteristics interact with algorithmic behavior.

Theoretical Complexity of the Evaluated Algorithms

The theoretical time and space complexities of the four algorithms are summarized in Table 2. Big-O notation describes the upper bound on the growth rate of resource consumption as a function of input size *n*, providing a hardware-independent reference against which empirical observations can be compared [13], [14].

Table 2. Theoretical Complexity of the Evaluated Search Algorithms

Algorithm	Time Complexity	Space Complexity	Required Precondition
Baseline linear search	*O(n)*	*O(1)*	NoneNone
Binary search	*O(log n)*	*O(1)*	Sorted dataset
Tree-based search	*O(log n)*	*O(n)*	Indexed structure
N-gram search	*O(n · k)*	*O(n · k)*	Token index

n=number of lemmas; k=number of n-gram tokens per lemma.
Source: (Research Results, 2023)

Linear search must inspect each record and therefore scales linearly with *n*. Binary search exploits alphabetical ordering to halve the search space at each step, achieving logarithmic time but requiring sorted input. Tree-based search achieves comparable logarithmic time through hierarchical indexing, at the cost of additional space for the index structure [15]. N-gram search introduces both time and space overhead proportional to the number of generated substring tokens, in exchange for partial-match capability.

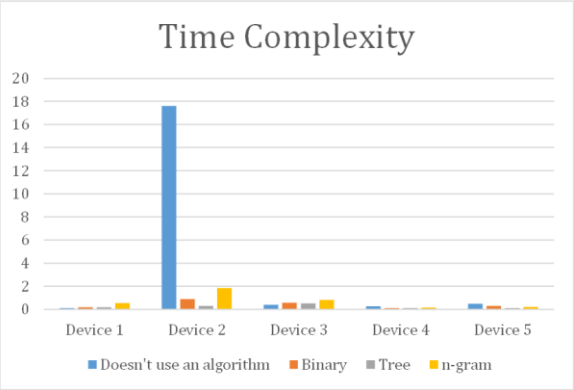
Empirical Time Complexity

Table 3 and Figure 6 report the search time per algorithm across the five testing devices.

Table 3. Time Complexity

Devices name	Baseline linear search without optimization	Binary	Tree	n-gram	Σ
Device 1	0,11	0,19	0,18	0,55	1,03
Device 2	17,61	0,89	0,31	1,83	20,64
Device 3	0,4	0,58	0,53	0,82	2,33
Device 4	0,28	0,13	0,13	0,14	0,68
Device 5	0,48	0,3	0,13	0,21	1,12
Σ	18,88	2,09	**1,28**	3,55	

Source : (Research Results, 2023)



Source : (Research Results, 2023)
Figure 6. Time Complexity Comparison Graph

Across all devices, the empirical search times are broadly consistent with the theoretical predictions in Table 2. Tree-based search achieves the lowest cumulative search time (1.28 s) and the lowest per-device time on four of the five devices, in line with its *O(log n)* time complexity and index-supported lookup. Binary search follows (2.09 s cumulative), also consistent with *O(log n)*. N-gram search exhibits higher search times (3.55 s cumulative) due to its *O(n · k)* complexity, which grows with the number of substring tokens. The unoptimized baseline accumulates the highest cumulative search time (18.88 s), as expected from its *O(n)* characteristic. This trend mirrors findings reported on large-scale corpora, where algorithms with sub-linear complexity consistently outperform linear search as data size grows [28].

A notable anomaly appears on Device 2, where the baseline linear search required 17.61 seconds, an order of magnitude higher than on the other devices. Three factors plausibly contribute to this observation.

First, Big-O notation hides multiplicative constants and lower-order terms that, in practice, materially affect runtime [14]. Device 2 (Intel Core i3 gen 7, 12 GB RAM, 128 GB SSD) has the weakest processor and the smallest storage among the five devices. On linear search, where every lemma record must be inspected, these constant factors compound across all 5,597 iterations, magnifying differences that are invisible in pure Big-O analysis.

Second, the baseline operates without any auxiliary index, forcing the database engine to perform a full table scan for every query. On devices with slower I/O subsystems, this disproportionately penalizes the *O(n)* algorithm relative to *O(log n)* algorithms, which traverse only a small fraction of the data. Crucially, this anomaly affects only the baseline; on the same Device 2, binary (0.89 s), tree-based (0.31 s), and n-gram (1.83 s) algorithms all remain within reasonable

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.5948

bounds, indicating that the anomaly is algorithm-specific rather than a general device failure.

The Device 2 result therefore reinforces, rather than contradicts, the theoretical analysis: algorithms with poor asymptotic complexity (*O(n)*) are precisely those most vulnerable to hardware-dependent slowdown on constrained devices [21], exactly the condition under which Tourism 4.0 applications must remain responsive.

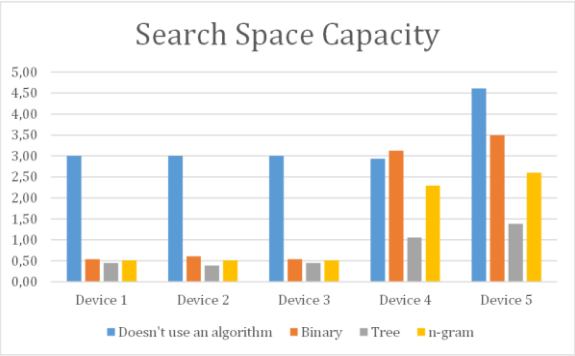
Empirical Space Complexity

Table 4 and Figure 7 report memory consumption per algorithm across devices.

Table 4. Search space capacity (memory)

Devices name	Baseline linear search without optimization	Binary	Tree	n-gram	Σ
Device 1	3,00	0,54	0,45	0,51	4,50
Device 2	3,00	0,61	0,39	0,51	4,51
Device 3	3,00	0,54	0,45	0,51	4,50
Device 4	2,93	3,13	1,06	2,29	9,41
Device 5	4,61	3,50	1,38	2,60	12,09
Σ	16,54	8,32	**3,73**	6,42	

Source : (Research Results, 2023)



Source : (Research Results, 2023)
Figure 7. Search Space Capacity Comparison Graph

Tree-based search records the lowest cumulative memory usage (3.73 kB) and the smallest per-device usage on four of the five devices, consistent with leveraging an existing database-level index rather than constructing additional in-memory structures during search. Binary search follows (8.32 kB), reflecting *O(1)* search-routine behavior with PHP runtime overhead. N-gram search (6.42 kB) consumes more memory than binary due to additional token structures required for substring matching, in line with its $O(n \cdot k)$ space complexity. The unoptimized baseline records the highest cumulative memory usage (16.54 kB), driven by loading large result sets into memory during sequential scanning.

Memory measurements on Devices 4 (MacBook Pro M1) and Device 5 (MacBook Air) are systematically higher than on Devices 1–3, likely reflecting differences in PHP runtime and operating-system environments between macOS and Windows, where memory-allocation granularity and PHP's internal buffer management differ. Despite these environmental differences, the *relative ranking* of algorithms remains consistent across all five devices, supporting the conclusion that the observed differences reflect intrinsic algorithmic behavior.

Implications for Algorithm Selection

Synthesizing the empirical and theoretical results, the tree-based search algorithm emerges as the most suitable choice for the Papuan regional-language digital dictionary. It combines the strongest empirical performance (lowest mean search time of 1.28 s and lowest mean memory usage of 3.73 kB) with favorable theoretical properties (*O(log n)* time complexity with index-supported lookup). Its performance also remains stable across heterogeneous devices, which is critical for Tourism 4.0 deployment scenarios where end-user devices vary substantially in computational capacity [19].

Despite its superior performance, further optimization remains possible. Trie-based or prefix-tree algorithms can be investigated to optimize prefix-search operations common in dictionary applications [29]. Hybrid approaches combining tree-based search with n-gram indexing may balance retrieval accuracy and memory efficiency, particularly for handling regional-language spelling variations. Approximate string-matching techniques, such as edit distance, can also be integrated to improve user experience under typographical errors. Scalability testing on larger lexical datasets would further support deployment in high-frequency-access Tourism 4.0 environments.

CONCLUSION

This study examined the computational complexity of search algorithms in a digital dictionary for Papuan regional languages, a low-resource language context relevant to Tourism 4.0 deployment. Three algorithms (binary search, tree-based search, and n-gram search) were benchmarked against an unoptimized linear search baseline using a real dataset of 5,597 lemmas across five heterogeneous computing devices, with

evaluation grounded in both Big-O theoretical analysis and empirical measurement of time and memory consumption.

Three main findings emerge. First, the unoptimized baseline, which relies on linear search, exhibits the highest time and space complexity, making it increasingly inefficient as the lexical corpus grows. Second, the experimental results show that the tree-based search algorithm achieves the best overall performance across heterogeneous devices, with the lowest average search time of 1.28 seconds and the smallest average memory usage of 3.73 kB. Third, hardware specifications, particularly processor capability and storage type, significantly influence absolute search performance, although the relative ranking of algorithms remains consistent, confirming that algorithm choice rather than hardware variability is the dominant factor in determining search efficiency.

The empirical patterns are consistent with the theoretical Big-O predictions, supporting the suitability of tree-based search for regional-language digital dictionaries deployed under the device heterogeneity that characterizes Tourism 4.0 ecosystems. By providing both fast information access and minimal memory consumption, the recommended algorithm contributes to the digital infrastructure required for preservation, learning, and tourism-related use of Papuan regional languages.

This study has several limitations. Each $(algorithm \times query\ word \times testing\ devices)$ combination was measured once, which may be susceptible to transient system noise; future work will adopt multi-run averaging with statistical analysis to strengthen measurement robustness. Additionally, the evaluation employed a limited lexical dataset and a restricted set of algorithms, which may limit generalizability to larger corpora or to other regional languages with different linguistic characteristics. Future research may explore trie-based structures, hybrid tree-and-n-gram approaches, or approximate string-matching techniques to further optimize search accuracy and performance in regional-language digital dictionaries.

ACKNOWLEDGEMENTS

This research was funded by DRTPM of the Ministry of Education, Culture, Research, and Technology based on Decree No. 190/E5/PG.02.00.PL/2023 dated 19 June 2023 through the Beginner Lecturer Research (PDP) grant scheme for the 2023 fiscal year. The author would like to thank the DRTPM Directorate General of Education and Technology, LLDIKTI Region XIV, LPPM Universitas Muhammadiyah Papua, and Balai Bahasa Provinsi Papua.

REFERENCE

[1] N. F. Hasan, A. Aisyah, R. Rahman, and H. Wonda, "Sentiment Analysis of Public Opinion Regarding Papuan Local Languages Condition Using Data Science Approach," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 13, no. 2 SE-Articles, pp. 125–139, Nov. 2022, doi: 10.31849/digitalzone.v13i2.11545.

[2] J. Kusuma, "Revolusi industri 4.0 memperkuat keberagaman bahasa daerah," no. 1, pp. 103–106, 2022, [Online]. Available: http://sinarbahtera.kemdikbud.go.id/index.php/SB/article/view/88

[3] A. F. Aji *et al.*, "One Country, 700+ Languages: NLP Challenges for Underrepresented Languages and Dialects in Indonesia," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, Dublin, Ireland: Association for Computational Linguistics, pp. 7226–7249. doi: 10.18653/v1/2022.acl-long.500.

[4] J. Mothe, "Shaping the Future of Endangered and Low-Resource Languages—Our Role in the Age of LLMs: A Keynote at ECIR 2024," *arXiv Prepr.*, vol. 58, no. 1, pp. 1–13, 2024.

[5] G. B. Dalkiran, "The Effects of Industry 4.0 Components on the Tourism Sector," 2022, pp. 235–250. doi: 10.1007/978-981-16-5644-6_14.

[6] A. Jariah, "Implementasi Literasi Digital Dalam Peningkatan Daya Tarik Wisata Di Era New Normal Kota Palangka Raya," *J. Hadratul Madaniyah*, vol. 8, no. 1, pp. 74–87, 2021, doi: 10.33084/jhm.v8i1.2441.

[7] V. Rodrigues, Z. Breda, and C. Rodrigues, "The Implications of Industry 4.0 for The Tourism Sector: A systematic literature review," *Heliyon*, vol. 10, no. 1, 2024, doi: 10.1016/j.heliyon.2024.e31590.

[8] W. Wu, C. Xu, M. Zhao, X. Li, and R. Law, "Digital Tourism and Smart Development: State-of-the-Art Review," *Sustainability*, vol. 16, no. 23, pp. 1–19, 2024, doi: 10.3390/su162310382.

[9] A. K. Sharma and R. Sharma, "Smart tourism in the digital age: overcoming barriers and unlocking," *Rev. Gestão*, vol. 32, no. 3, pp. 224–237, 2025, doi: 10.1108/REGE-02-2025-0030.

[10] K. Kasmawati, B. Kadir, A. Agussalim, R.

VOL. 11. NO. 4 MAY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i4.5948

Renold, F. Firmansyah, and M. Antasari, "The Use of Digital Library for Tourism Promotion Through Tourism Object Transliteration into Katakana and Lontara Letter," *Pusaka J. Tour. Hosp. Travel Bus. Event*, vol. 5, no. 1, pp. 11–19, 2023, doi: 10.33649/pusaka.v5i1.190.

[11] N. F. Hasan and M. S. A. Iribaram, "Digitizing the Papuan Regional Language Dictionary using the Rapid Application Development Method," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 3, pp. 710–720, 2022, doi: 10.30812/matrik.v21i3.1688.

[12] Y. M. Mantri, "Digitalisasi Bahasa Daerah Sebagai Upaya Meningkatkan Ketahanan Budaya Daerah," *Textura J.*, vol. 2, no. 2, pp. 67–83, 2021, [Online]. Available: http://journal.piksi.ac.id/index.php/TEXTURA

[13] S. Phalke, Y. Vaidya, and S. Metkar, "Big-O Time Complexity Analysis Of Algorithm," in *2022 International Conference on Signal and Information Processing (IConSIP)*, 2022, pp. 1–5. doi: 10.1109/ICoNSIP49665.2022.10007469.

[14] F. A. Mala and R. Ali, "The Big-O of Mathematics and Computer Science," *J. Appl. Math. Comput.*, vol. 6, no. 1, pp. 1–3, 2022, doi: 10.26855/jamc.2022.03.001.

[15] J. Singh, "Computational Complexity and Analysis of Supervised Machine Learning Algorithms BT - Next Generation of Internet of Things," R. Kumar, P. K. Pattnaik, and J. M. R. S. Tavares, Eds., Singapore: Springer Nature Singapore, 2023, pp. 195–206.

[16] 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, pp. 75–80, 2021, doi: 10.30646/tikomsin.v9i1.546.

[17] N. Arifin, F. Fauziah, and N. Nurhayati, "Kombinasi Algoritma Sequential Searching dan Bubble Sort Pada Manajemen Laboratorium," *J. Sains Komput. Inform.*, vol. 6, no. 1, pp. 294–306, 2022, [Online]. Available: http://www.tunasbangsa.ac.id/ejurnal/index.php/jsakti/article/view/445

[18] W. S. Wahyuni, S. Andryana, and B. Rahman, "Penggunaan Algoritma Sequential Searching Pada Aplikasi Perpustakaan Berbasis Web," *J. Ilm. Penelit. dan Pembelajaran Inform.*, vol. 07, no. 02, pp. 294–302, 2022, [Online]. Available: https://www.jurnal.stkippgritulungagung.ac.id/index.php/jipi/article/view/2646

[19] Z. Zhang, "Computational Technologies for Construction of Business Korean Translation Corpus Based on Association Rules Mining," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, 2023, doi: 10.1145/3603500.

[20] I. Arapakis, S. Park, and M. Pielot, "Impact of Response Latency on User Behaviour in Mobile Web Search," in *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval (CHIIR '21)*, Canberra, Australia: ACM, pp. 279–283. doi: 10.1145/3406522.3446038.

[21] V. J. Reddi *et al.*, "MLPerf Mobile Inference Benchmark," in *Proceedings ofthe 5th Machine Learning and Systems (MLSys) Conference*, Santa Clara, CA, USA: Cornell University, 2022, pp. 1–18. doi: https://doi.org/10.48550/arXiv.2012.02328.

[22] N. Q. Dhaneswara, K. Nasution, and T. Haramaini, "Perancangan Aplikasi Kamus Digital Bahasa Minang dengan Menggunakan Metode String Matching Knuth Morris Praat," *J. Minfo Polgan*, vol. 10, no. 1, pp. 1–7, 2021, doi: 10.33395/jmp.v10i1.11018.

[23] R. Rismayani, N. Sambo Layuk, S. Wahyuni, H. Wali, and N. K. Marselina, "Pencarian Kata Pada Aplikasi Kamus Istilah Komputer dan Informatika Menggunakan Algoritma Brute Force Berbasis Android," *Komputika J. Sist. Komput.*, vol. 10, no. 1, pp. 43–52, 2021, doi: 10.34010/komputika.v10i1.3644.

[24] 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, pp. 21–30, 2021, [Online]. Available: https://ojs.uajy.ac.id/index.php/jbi/article/view/4367

[25] D. J. Zalukhu, P. Karo-karo, and N. M. Faizah, "Perancangan Aplikasi Kamus Bahasa Daerah Nias Berbasis Android dengan Metode Rapid Application Development (RAD) Menggunakan Android Studio," *Comput. J.*, vol. 1, no. 1, pp. 9–14, 2023, doi: 10.58477/cj.v1i1.30.

[26] R. Y. Darmawantoro, Y. R. W. Utami, and K. Kustanto, "Implementasi Binary Search Untuk Data Obat di Apotek," *J. Teknol. Inf. dan Komun.*, vol. 10, no. 1, 2022, doi: 10.30646/tikomsin.v10i1.607.

[27] R. Rismayani, N. Sambo Layuk, S. Wahyuni,

H. Wali, and N. K. Marselina, "Pencarian Kata Pada Aplikasi Kamus Istilah Komputer dan Informatika Menggunakan Algoritma Brute Force Berbasis Android," *Komputika J. Sist. Komput.*, vol. 10, no. 1, pp. 43–52, 2021, doi: 10.34010/komputika.v10i1.3644.

[28] N. Purnama, "Comparative Performance Study of Search Algorithms on Large-Scale Data Structures," *J. Ilmu Pengetah. dan Teknol. Komput.*, vol. 11, no. 1, pp. 99–109, 2025, doi: 10.33480/jitk.v11i1.

[29] N. R. Feta and F. Fitria, "Implementation of Concolic Unit Testing in Testing Binary Search Algorithm Using Jcute," *JITK (Jurnal Ilmu Pengetah. dan Teknol. Komputer)*, vol. 7, no. 2, pp. 37–44, 2022, doi: 10.33480/jitk.v7i2.2758.

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri