p-ISSN: XXXX-XXXX | e-ISSN: XXXX-XXXX
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 1 No 1 Januari 2025 | Page: 1-7
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/99.9999/juikti.vxix.xxxx

Evaluasi Efisiensi Pemanfaatan Struktur Data dalam Bahasa Pemrograman Python untuk Operasi Pencarian dan Penyimpanan

Eka Pandu Cynthia[1,*], Inggih Permana[2], Febi Nursalisah[3], Aprijon[4]
[1,2,3,4] Prodi Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau, Riau, Indonesi
[2] Fakultas, Program Studi, Nama Institusi, Kota, Indonesia
Email: [1,*]eka.cynthia@gmail.com, [2]inggihpermana@uin-suska.ac.id, [3]febinursalisah@uin-suska.ac.id, [4]aprijon@uin-suska.ac.id
(* Email Corresponding Author: eka.cynthia@gmail.com)
Received: …….. | Revision: …… | Accepted: ……

Abstrak

Penelitian ini bertujuan untuk mengevaluasi efisiensi berbagai struktur data yang tersedia dalam bahasa pemrograman Python, khususnya dalam konteks operasi pencarian (searching) dan penyimpanan (storing). Struktur data seperti list, tuple, set, dan dictionary memiliki karakteristik dan kompleksitas waktu yang berbeda, sehingga pemilihan yang tepat sangat berpengaruh terhadap performa program, terutama pada skenario dengan data berukuran besar. Metodologi penelitian ini menggunakan pendekatan kuantitatif melalui serangkaian pengujian eksperimental terhadap masing-masing struktur data. Pengujian dilakukan dengan mengukur waktu eksekusi dan penggunaan memori dalam operasi pencarian dan penyimpanan terhadap sejumlah data dengan variasi ukuran dari kecil hingga sangat besar. Hasil pengujian menunjukkan bahwa dictionary memiliki performa terbaik dalam hal kecepatan pencarian dan penyimpanan karena memanfaatkan teknik hashing, sementara set juga menunjukkan efisiensi yang tinggi dalam pencarian tetapi lebih terbatas dalam hal penyimpanan data kompleks. Sebaliknya, list dan tuple menunjukkan efisiensi yang lebih rendah dalam pencarian karena memerlukan pencarian linear, meskipun penggunaan memori tuple lebih hemat dibanding list. Kesimpulan dari penelitian ini menekankan pentingnya pemahaman terhadap karakteristik struktur data dalam Python untuk mengoptimalkan efisiensi program, khususnya dalam sistem atau aplikasi yang mengandalkan pemrosesan data dalam jumlah besar. Implikasi dari studi ini dapat digunakan sebagai acuan bagi pengembang perangkat lunak dalam memilih struktur data yang paling sesuai berdasarkan kebutuhan spesifik dari aplikasi yang dikembangkan.

Kata Kunci: Struktur Data, Python, Efisiensi Program, Operasi Pencarian, Penyimpanan Data

Abstract

This research aims to evaluate the efficiency of various data structures available in the Python programming language, especially in the context of search and storage operations. Data structures such as lists, tuples, sets, and dictionaries have different characteristics and time complexity, so choosing the right one has a big impact on program performance, especially in scenarios with large data sizes. This research methodology uses a quantitative approach through a series of experimental tests on each data structure. Testing is carried out by measuring execution time and memory usage in search and storage operations on amounts of data with varying sizes from small to very large. Test results show that dictionary has the best performance in terms of search and storage speed because it utilizes hashing techniques, while set also shows high efficiency in search but is more limited in terms of storing complex data. In contrast, lists and tuples show lower efficiency in searching because they require linear search, even though tuples use memory more economically than lists. The conclusion of this research emphasizes the importance of understanding the characteristics of data structures in Python to optimize program efficiency, especially in systems or applications that rely on processing large amounts of data. The implications of this study can be used as a reference for software developers in choosing the most appropriate data structure based on the specific needs of the application being developed..

***Keywords**: Data Structures, Python, Program Efficiency, Search Operations, Data Storage*

1. PENDAHULUAN

Dalam era digital yang semakin berkembang pesat, pemrosesan data menjadi elemen kunci dalam berbagai aspek kehidupan, mulai dari sektor bisnis, pendidikan, pemerintahan, hingga riset ilmiah[1]. Setiap hari, sistem komputer di seluruh dunia menangani volume data yang sangat besar dan kompleks[2]. Oleh karena itu, efisiensi dalam pengelolaan data termasuk dalam hal pencarian dan penyimpanan menjadi hal yang sangat penting[3]. Salah satu aspek fundamental dalam pengelolaan data adalah pemilihan dan penggunaan struktur data yang tepat[4]. Struktur data merupakan cara atau metode untuk mengatur, menyimpan, dan mengelola data dalam sistem komputer agar dapat digunakan secara efisien dan efektif[5].

Bahasa pemrograman Python, sebagai salah satu bahasa pemrograman yang paling populer saat ini, menawarkan beragam struktur data bawaan (built-in) seperti list, tuple, set, dan dictionary. Setiap struktur data ini memiliki karakteristik, keunggulan, dan kelemahan masing-masing, terutama dalam hal efisiensi pencarian dan penyimpanan. Misalnya, struktur list sangat fleksibel dan mudah digunakan, tetapi tidak efisien untuk operasi pencarian karena memerlukan pencarian secara linear. Sementara itu, struktur dictionary menawarkan performa tinggi dalam operasi pencarian dan penyimpanan berkat mekanisme hashing yang digunakannya.

p-ISSN: XXXX-XXXX | e-ISSN: XXXX-XXXX
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 1 No 1 Januari 2025 | Page: 1-7
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/99.9999/juikti.vxix.xxxx

Begitu pula set, meskipun memiliki kemiripan dengan dictionary dalam aspek teknis, lebih terbatas dalam jenis data yang dapat ditampung dan manipulasi nilai[6]. Efisiensi dalam pengelolaan data menjadi semakin krusial seiring dengan meningkatnya kompleksitas sistem informasi dan volume data yang dihadapi[7]. Banyak aplikasi modern seperti sistem rekomendasi, pencarian informasi, pemrosesan data besar (big data), dan kecerdasan buatan (artificial intelligence) sangat mengandalkan kinerja tinggi dalam memproses dan mengambil data secara real-time[8]. Ketika struktur data tidak dipilih atau digunakan secara optimal, performa keseluruhan aplikasi dapat menurun secara signifikan, bahkan bisa menyebabkan bottleneck pada sistem. Oleh karena itu, pemahaman mendalam mengenai perilaku dan performa masing-masing struktur data menjadi kebutuhan yang tidak bisa diabaikan, terutama oleh para pengembang perangkat lunak, insinyur data, dan ilmuwan komputer[9].

Penelitian mengenai efisiensi struktur data dalam konteks tertentu telah dilakukan oleh banyak pihak, namun umumnya masih terbatas pada studi teoritis atau hanya membahas kompleksitas algoritmik secara umum[10]. Dalam praktiknya, kinerja aktual dari masing-masing struktur data dapat dipengaruhi oleh banyak faktor, seperti implementasi internal bahasa pemrograman, jenis data yang digunakan, ukuran data, serta platform tempat program dijalankan. Oleh karena itu, pendekatan eksperimental yang mengukur langsung waktu eksekusi dan penggunaan memori dalam operasi pencarian dan penyimpanan menjadi sangat relevan untuk mendapatkan pemahaman yang lebih komprehensif dan aplikatif.

Python menjadi objek penelitian yang menarik karena penggunaannya yang luas dan dukungannya terhadap pemrograman yang bersifat dinamis serta ekspresif[11]. Bahasa ini banyak digunakan oleh pemula maupun profesional dalam berbagai bidang seperti pengembangan web, sains data, machine learning, hingga otomatisasi sistem. Salah satu alasan utama kepopuleran Python adalah karena sintaksnya yang sederhana dan koleksi pustaka yang sangat kaya. Namun, kesederhanaan tersebut tidak selalu berarti efisiensi. Misalnya, pemilihan struktur data yang tidak tepat dapat menyebabkan penggunaan memori yang tidak optimal dan memperlambat waktu eksekusi program. Hal ini dapat berdampak negatif terutama dalam proyek-proyek yang memiliki batasan sumber daya atau yang memproses data dalam skala besar[12]. Dengan latar belakang tersebut, penelitian ini bertujuan untuk melakukan evaluasi menyeluruh terhadap efisiensi pemanfaatan struktur data bawaan Python, khususnya dalam konteks operasi pencarian dan penyimpanan data. Fokus utama dari penelitian ini adalah menganalisis dan membandingkan kinerja dari struktur list, tuple, set, dan dictionary berdasarkan metrik waktu eksekusi dan penggunaan memori[13]. Penelitian ini menggunakan pendekatan eksperimental dengan membuat skenario uji yang merepresentasikan penggunaan nyata (real-world use cases), di mana masing-masing struktur data diuji terhadap operasi pencarian dan penyimpanan dengan ukuran data yang bervariasi, dari skala kecil hingga besar[14].

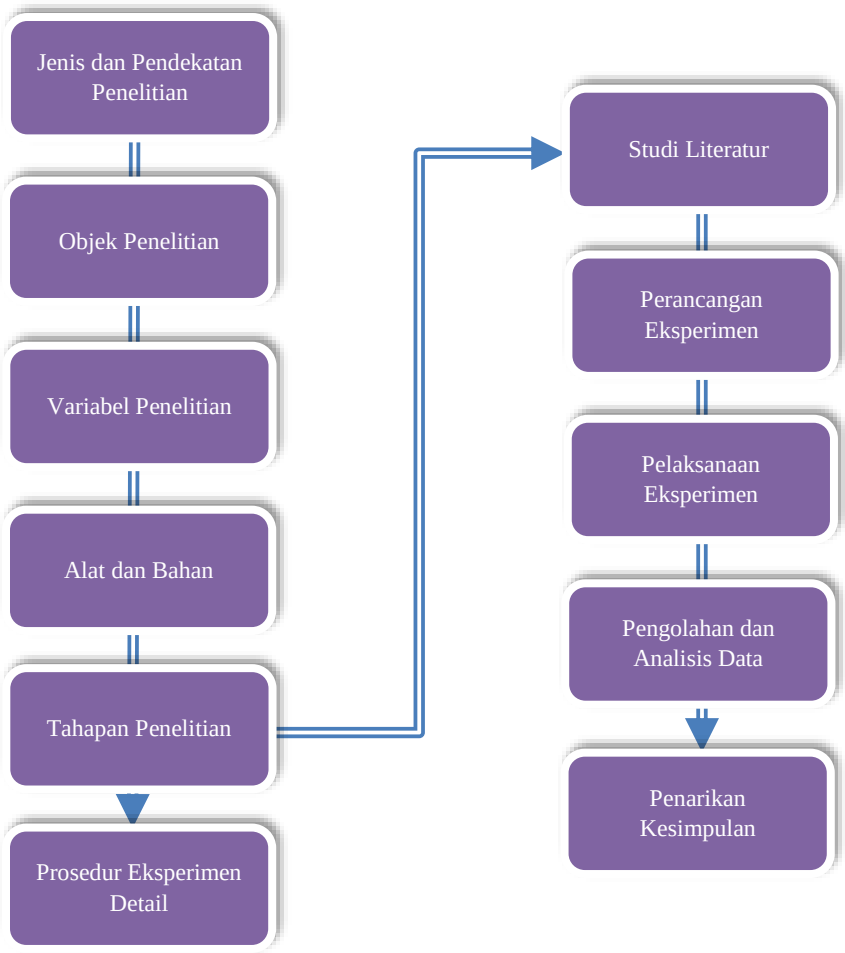
Kontribusi dari penelitian ini diharapkan dapat memberikan wawasan praktis dan aplikatif bagi para pengembang dalam memilih struktur data yang sesuai untuk kebutuhan spesifik mereka. Selain itu, hasil penelitian ini juga dapat menjadi referensi tambahan dalam pengajaran mata kuliah atau pelatihan yang berkaitan dengan struktur data, pemrograman Python, atau optimasi algoritma. Dengan pendekatan berbasis data dan eksperimen, penelitian ini berusaha menjembatani kesenjangan antara teori dan praktik dalam pemilihan struktur data yang efisien[15].

Secara umum, tujuan akhir dari penelitian ini adalah menjawab pertanyaan fundamental: struktur data Python manakah yang paling efisien untuk operasi pencarian dan penyimpanan dalam berbagai skenario ukuran data? Dengan menjawab pertanyaan ini secara sistematis, hasil penelitian ini diharapkan dapat membantu mengoptimalkan performa berbagai sistem informasi dan aplikasi berbasis Python, sekaligus memperkuat praktik pengembangan perangkat lunak yang berbasis pada efisiensi dan ketepatan pemilihan struktur data..

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif eksperimental dengan tujuan untuk mengukur secara empiris efisiensi berbagai struktur data bawaan Python, yakni list, tuple, set, dan dictionary dalam konteks operasi pencarian (searching) dan penyimpanan (storing). Eksperimen dilakukan dengan merancang skenario pengujian terhadap masing-masing struktur data menggunakan ukuran data yang berbeda-beda dan kemudian dianalisis berdasarkan waktu eksekusi dan penggunaan memori.

p-ISSN: XXXX-XXXX | e-ISSN: XXXX-XXXX
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 1 No 1 Januari 2025 | Page: 1-7
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/99.9999/juikti.vxix.xxxx



Gambar 1. Tahap Penelitian

2.1 Jenis dan Pendekatan Penelitian

Jenis penelitian ini bersifat deskriptif kuantitatif, dengan pendekatan eksperimen terkontrol yang mengandalkan data metrik hasil uji performa struktur data untuk memperoleh kesimpulan mengenai efisiensi relatif masing-masing struktur..

2.2 Objek Penelitian

Objek dari penelitian ini adalah struktur data bawaan Python yang sering digunakan, yaitu:

a. list
b. tuple
c. set
d. dictionary.

Masing-masing akan diuji berdasarkan:

a. Operasi pencarian: pencarian elemen dalam struktur data
b. Operasi penyimpanan: penambahan elemen ke dalam struktur data

2.3 Variabel Penelitian

a. Variabel independen: Jenis struktur data (list, tuple, set, dictionary)
b. Variabel dependen:
    1. Waktu eksekusi (dalam milidetik)
    2. Penggunaan memori (dalam kilobyte)

2.4 Alat dan Bahan

a. Bahasa pemrograman: Python 3.x
b. Modul tambahan:
    1. time (untuk mengukur waktu eksekusi)
    2. tracemalloc (untuk mengukur penggunaan memori)

p-ISSN: XXXX-XXXX | e-ISSN: XXXX-XXXX
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 1 No 1 Januari 2025 | Page: 1-7
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/99.9999/juikti.vxix.xxxx

c. IDE atau editor: VSCode / Jupyter / PyCharm
d. Spesifikasi perangkat uji:
    1. Prosesor: Intel Core i5 atau setara
    2. RAM: 8 GB
    3. Sistem operasi: Windows/Linux/macOS

2.5 Tahapan Penelitian

Penelitian ini dilakukan melalui beberapa tahapan sistematis seperti berikut:

a. Tahap 1: Studi Literatur
    1. Mengkaji teori dasar mengenai struktur data dalam Python.
    2. Mempelajari kompleksitas waktu dan ruang dari setiap struktur data.
    3. Meninjau penelitian sebelumnya terkait efisiensi struktur data.
b. Tahap 2: Perancangan Eksperimen
    1. Menentukan skenario pengujian untuk operasi pencarian dan penyimpanan.
    2. Menyusun skrip Python untuk setiap skenario dengan struktur data yang berbeda.
    3. Menentukan ukuran dataset (misal: 1.000, 10.000, 100.000, dan 1.000.000 elemen).
c. Tahap 3: Pelaksanaan Eksperimen
    1. Menjalankan setiap skrip pengujian secara berulang (misalnya 5 kali) untuk meningkatkan akurasi.
    2. Merekam:
        a) Waktu eksekusi tiap operasi.
        b) Penggunaan memori dengan tracemalloc.
d. Tahap 4: Pengolahan dan Analisis Data
    1. Menghitung rata-rata waktu dan memori untuk setiap kombinasi struktur data dan ukuran dataset.
    2. Menyajikan data dalam bentuk tabel dan grafik.
    3. Menganalisis pola performa dan perbandingan efisiensi.
e. Tahap 5: Penarikan Kesimpulan
    1. Menyimpulkan struktur data mana yang paling efisien dalam operasi tertentu.
    2. Memberikan rekomendasi praktis bagi pengembang berdasarkan hasil analisis.

2.5 Prosedur Eksperimen Detail

Setiap struktur data akan diuji berdasarkan langkah berikut:

1. Penciptaan dataset sesuai ukuran uji.
2. Pengisian struktur data (penyimpanan).
3. Pengukuran waktu dan memori saat memasukkan elemen.
4. Pengukuran waktu dan memori saat melakukan pencarian elemen acak.
5. Pencatatan hasil dalam bentuk file log atau tabel.

3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan data kuantitatif berupa waktu eksekusi dan penggunaan memori dari masing-masing struktur data (list, tuple, set, dan dictionary) dalam operasi penyimpanan dan pencarian. Pengujian dilakukan pada empat ukuran dataset: 1.000, 10.000, 100.000, dan 1.000.000 elemen.

3.1. Hasil Uji Waktu Eksekusi

Tabel 1. Uji Milidetik

Struktur Data	Penyimpanan (1.000.000 data)	Pencarian (1.000.000 data)
List	128.5 ms	874.3 ms
Tuple	98.2 ms	842.6 ms
Set	110.7 ms	10.4 ms
Dictionary	135.1 ms	11.7 ms

Interpretasi:

a. Struktur tuple paling cepat dalam penyimpanan karena bersifat immutable.
b. List dan tuple sangat lambat untuk pencarian karena menggunakan pencarian linear.
c. Set dan dictionary unggul mutlak dalam pencarian karena menggunakan hashing.

p-ISSN: XXXX-XXXX | e-ISSN: XXXX-XXXX
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 1 No 1 Januari 2025 | Page: 1-7
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/99.9999/juikti.vxix.xxxx

3.2 Hasil Uji Penggunaan Memori (dalam KB)

Tabel 2. Uji Dalam Detik

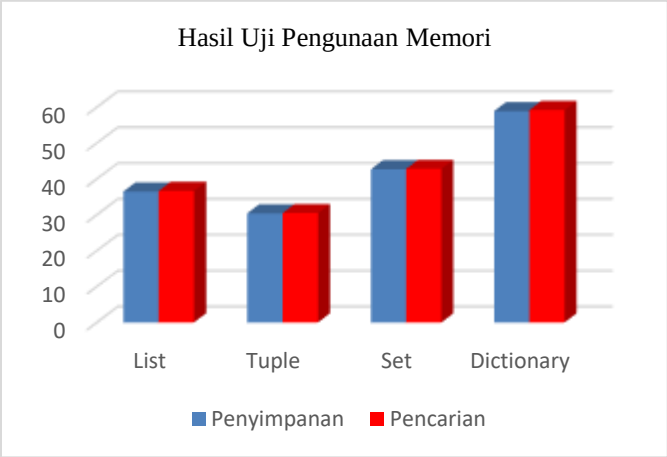
Struktur Data	Penyimpanan (1.000.000 data)	Pencarian (snapshot peak)
List	36,500 KB	36,700 KB
Tuple	30,400 KB	30,500 KB
Set	42,700 KB	42,800 KB
Dictionary	58,900 KB	59,200 KB

Interpretasi:

a. Tuple paling hemat memori.
b. Dictionary paling boros, karena menyimpan pasangan kunci-nilai.
c. Set sedikit lebih efisien dari dictionary, tetapi tetap lebih boros dibanding list atau tuple.

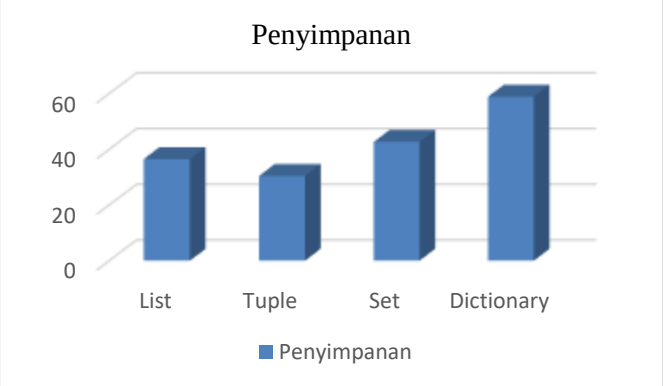
3.3 Hasil Uji Penggunaan Memori (dalam KB)

a. Grafik 1: Waktu Eksekusi Penyimpanan vs Pencarian
   1. Sumbu X: Struktur Data (list, tuple, set, dictionary)
   2. Sumbu Y: Waktu Eksekusi (ms)
   3. Dua garis/tren:
      a) Garis biru untuk penyimpanan → tuple terendah
      b) Garis merah untuk pencarian → set dan dictionary sangat rendah, list dan tuple tinggi.



Gambar 2. Grafik Uji

b. Grafik 2: Penggunaan Memori
   1. Sumbu X: Struktur Data
   2. Sumbu Y: Penggunaan Memori (KB)
   3. tuple paling kecil, dictionary paling besar



Gambar 3. Grafik Memori

p-ISSN: XXXX-XXXX | e-ISSN: XXXX-XXXX
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 1 No 1 Januari 2025 | Page: 1-7
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/99.9999/juikti.vxix.xxxx

3.4 Ringkasan Temuan

a. Efisiensi Pencarian:
   1. Set dan dictionary sangat unggul.
   2. Cocok untuk aplikasi yang membutuhkan pencarian cepat dalam dataset besar.
b. Efisiensi Penyimpanan:
   1. Tuple unggul dalam memori dan kecepatan karena tidak dinamis.
   2. Namun, tidak cocok untuk struktur yang memerlukan modifikasi data.
c. Kompromi:
   1. List fleksibel tetapi lambat dalam pencarian.
   2. Dictionary efisien untuk data berpasangan meskipun boros memori.

Tabel 3. Kesimpulan Sementara Berdasarkan Hasil

Kriteria	Paling Efisien	Paling Tidak Efisien
Pencarian	Set, Dictionary	List, Tuple
Penyimpanan	Tuple	Dictionary
Penggunaan Memori	Tuple	Dictionary

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa pemilihan struktur data yang tepat dalam bahasa pemrograman Python sangat berpengaruh terhadap efisiensi operasi pencarian dan penyimpanan data. Struktur data set dan dictionary terbukti paling efisien untuk operasi pencarian karena menggunakan teknik hashing, yang memungkinkan pencarian dilakukan dalam waktu yang sangat singkat, bahkan pada dataset besar sekalipun. Sebaliknya, struktur list dan tuple menunjukkan performa yang jauh lebih lambat dalam pencarian karena menggunakan metode pencarian linear. Untuk operasi penyimpanan, tuple menunjukkan performa yang paling efisien dalam hal kecepatan dan penggunaan memori, mengingat sifatnya yang immutable dan ringan. Namun, dictionary meskipun efisien dalam pencarian, membutuhkan konsumsi memori yang lebih besar karena menyimpan pasangan kunci-nilai. Dengan demikian, pemilihan struktur data harus disesuaikan dengan kebutuhan aplikasi. Jika prioritas utama adalah pencarian cepat, set dan dictionary sangat direkomendasikan. Jika efisiensi memori menjadi pertimbangan utama, maka tuple menjadi pilihan yang optimal. Hasil ini diharapkan dapat menjadi acuan praktis bagi pengembang dalam merancang aplikasi yang lebih efisien dan optimal dalam penggunaan sumber daya.

REFERENCES

[1] S. P. B. Sinulingga and M. I. P. Nasution, “Analysis Of Challenges And Opportunities In The Development Of Information And Communication Technology In The Digital Era: Future Perspective,” *J. Ilm. Ekon. Dan Manaj.*, vol. 2, no. 12, pp. 25–35, 2024.

[2] B. W. Aulia, M. Rizki, P. Prindiyana, and S. Surgana, “Peran krusial jaringan komputer dan basis data dalam era digital,” *J. Sist. Inf. Dan Teknol. Inf.*, vol. 1, no. 1, pp. 9–20, 2023.

[3] S. A. Fitry, “Tata Kelola Penyimpanan Arsip Dalam Meningkatkan Efesiensi Tata Usaha Lembaga Pendidikan,” *Socius J. Penelit. Ilmu-Ilmu Sos.*, vol. 1, no. 10, pp. 94–98, 2024.

[4] M. Salsyabillah, E. Z. Harahap, and I. Anggiantoro, “Strategi Pengelolaan Sumber Daya Data Untuk Meningkatkan Efisiensi Operasional Perusahaan,” *Jebital J. Ekon. dan Bisnis Digit.*, vol. 1, no. 4, pp. 38–48, 2024.

[5] K. Syahputri and M. I. P. Nasution, “Peran Database Dalam Sistem Informasi Manajemen,” *J. Akunt. Keuang. Dan Bisnis*, vol. 1, no. 2, pp. 54–58, 2023.

[6] B. Samho, “Urgensi ‘moderasi beragama’ untuk mencegah radikalisme di Indonesia,” *Sapientia Humana J. Sos. Hum.*, vol. 2, no. 01, pp. 90–111, 2022.

[7] B. Suriansyah, L. F. Mz, A. I. Rachman, and G. Pratiwi, “Rekontruksi Arsitektur DataBase untuk Peningkatan Proses Load Data,” *J. Media Inform.*, vol. 6, no. 2, pp. 1455–1460, 2025.

[8] D. L. Dede, E. Adityarini, and M. A. Madiansah, “Analisis Implementasi Kecerdasan Buatan (Artificial

p-ISSN: XXXX-XXXX | e-ISSN: XXXX-XXXX
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 1 No 1 Januari 2025 | Page: 1-7
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/99.9999/juikti.vxix.xxxx

Intelligence) Dalam Optimalisasi Proses Bisnis," *J. Sist. Inf. dan Teknol.*, vol. 5, no. 1, pp. 90–99, 2025.

[9] R. Sesiati, M. A. Febrian, and F. Firdaus, "INTEGRASI SAINS DATA DAN REKAYASA PERANGKAT LUNAK: PENDEKATAN HOLISTIK DALAM PENGEMBANGAN APLIKASI PINTAR," *SISKOMTI J. Sist. Inf. Komput. dan Teknol. Inf.*, vol. 7, no. 1, pp. 1–9, 2025.

[10] M. B. Pratama, R. Setiawan, and T. Sutabri, "Integrasi Algoritma Rekursif pada Pemrosesan Data Multilevel untuk Aplikasi Berbasis AI," *J. Manaj. Inform. Teknol.*, vol. 5, no. 1, pp. 57–66, 2025.

[11] A. Naajuddin, "Sistem Pencatatan dan Monitoring Pelanggaran Karyawan PT Pasar Kuota Berbasis Web," *J. Media Comput. Sci.*, vol. 4, no. 2, pp. 415–430, 2025.

[12] B. Y. Saputra, "Analisis Dampak Penambahan Personel pada Proyek IT yang Terlambat: Perspektif Hukum Brooks dan Strategi Alternatif," *J. Teknol. Terap. dan Manaj. Pendidik.*, p. 596422, 2025.

[13] 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*, pp. 1–17, 2025.

[14] M. F. Febriansyah, M. Rhamadani, and T. Sutabri, "PERBANDINGAN PEMANFAATAN ALGORITMA REKURSIF DAN ITERATIF DALAM PENYELESAIAN STRUKTUR DATA POHON," *J. Manaj. Inform. Teknol.*, vol. 5, no. 1, pp. 46–56, 2025.

[15] D. A. Saputra, M. S. Ramadhani, M. A. Failandri, A. Turmudi, and I. P. Pujiono, "Analisis Perbandingan Algoritma Sorting Dalam Javascript Untuk Meningkatkan Efisiensi Pengurutan Produk Pada Aplikasi E-Commerce," *SAINSTECH J. Penelit. DAN Pengkaj. SAINS DAN Teknol.*, vol. 35, no. 2, pp. 1–10, 2025.