

## ANALISIS PERBANDINGAN DAN EVALUASI KINERJA BINARY TREE DAN HASH TABLE DALAM MENINGKATKAN EFISIENSI PENCARIAN DATA KOMPLEKS

Tendriani<sup>1</sup>, Adilah<sup>2</sup>, Fajriah<sup>3</sup>, Indah Aulia<sup>4</sup>, Afdal<sup>5</sup>, Rifki Musaid<sup>6</sup>, Ahmad Surgawan<sup>7</sup>, Ketrin Rinayanti Manullang<sup>8</sup>

[tenrianitenri98@gmail.com](mailto:tenrianitenri98@gmail.com)<sup>1</sup>, [adilahdilah830@gmail.com](mailto:adilahdilah830@gmail.com)<sup>2</sup>, [fajriahjia36@gmail.com](mailto:fajriahjia36@gmail.com)<sup>3</sup>, [indahhauliaa06@gmail.com](mailto:indahhauliaa06@gmail.com)<sup>4</sup>, [afdesign42@gmail.com](mailto:afdesign42@gmail.com)<sup>5</sup>, [kyrif12345@gmail.com](mailto:kyrif12345@gmail.com)<sup>6</sup>, [ahmadsurgawan20@gmail.com](mailto:ahmadsurgawan20@gmail.com)<sup>7</sup>, [ketrin.rm46@gmail.com](mailto:ketrin.rm46@gmail.com)<sup>8</sup>

Universitas Sulawesi Barat

### ABSTRAK

Pencarian data merupakan salah satu proses penting dalam sistem pengolahan data, terutama ketika jumlah dan kompleksitas data terus meningkat. Pemilihan struktur data yang tepat dapat memengaruhi kecepatan dan efisiensi proses pencarian. Penelitian ini bertujuan untuk menganalisis dan mengevaluasi perbandingan kinerja antara Binary Tree dan Hash Table dalam meningkatkan efisiensi pencarian data kompleks. Metode penelitian dilakukan melalui implementasi kedua struktur data menggunakan algoritma pencarian yang sesuai, kemudian dilakukan pengujian terhadap beberapa skenario data dengan jumlah dan tingkat kompleksitas yang berbeda. Parameter yang digunakan dalam evaluasi meliputi waktu pencarian, penggunaan memori, dan efisiensi proses pencarian. Hasil penelitian menunjukkan bahwa Hash Table memiliki keunggulan pada kecepatan pencarian data spesifik, sedangkan Binary Tree lebih unggul dalam pengelolaan data terurut dan pencarian rentang nilai. Penelitian ini diharapkan dapat memberikan rekomendasi penggunaan struktur data yang lebih efektif sesuai kebutuhan sistem dan karakteristik data yang digunakan.

**Kata Kunci:** Binary Tree, Hash Table, Algoritma Pencarian, Struktur Data, Efisiensi.

### PENDAHULUAN

Perkembangan teknologi informasi menyebabkan peningkatan jumlah data yang diproses dalam berbagai sistem komputer. Pencarian data menjadi salah satu proses penting karena hampir seluruh aplikasi memerlukan akses data secara cepat dan efisien. Kinerja proses pencarian dipengaruhi oleh metode dan struktur data yang digunakan.

Struktur data memiliki peranan penting dalam mengatur penyimpanan dan pengolahan data agar proses pencarian dapat dilakukan dengan lebih efektif. Dua struktur data yang umum digunakan untuk proses pencarian adalah Binary Tree dan Hash Table.

Hash Table merupakan struktur data yang menggunakan fungsi hash untuk memetakan data ke lokasi tertentu dalam memori sehingga memungkinkan pencarian dilakukan dengan waktu rata-rata sebesar  $O(1)$ . Sementara itu, Binary Tree menyusun data dalam bentuk hierarki yang memungkinkan proses pencarian dilakukan secara terstruktur dengan kompleksitas  $O(\log n)$  pada kondisi pohon yang seimbang.

Meskipun kedua metode tersebut memiliki tujuan yang sama, masing-masing memiliki karakteristik, kelebihan, dan kekurangan yang berbeda. Oleh karena itu, diperlukan penelitian untuk menganalisis dan membandingkan kinerja kedua struktur data tersebut dalam menangani pencarian data kompleks.

### METODOLOGI

Penelitian ini menggunakan metode eksperimen komparatif untuk menganalisis dan mengevaluasi kinerja Binary Tree dan Hash Table. Tahapan penelitian dilakukan melalui langkah berikut:

Identifikasi Masalah → Studi Literatur → Implementasi Algoritma → Pengujian Sistem → Pengumpulan Data → Analisis Hasil → Kesimpulan  
 Data yang digunakan terdiri dari:

- ID Data
- Nama
- Kategori
- Tanggal
- Informasi tambahan

Jumlah data yang diuji:

- 1.000 data
- 5.000 data
- 10.000 data
- 25.000 data
- 50.000 data

Variabel penelitian:

- Waktu pencarian (search time)
- Waktu penyisipan (insert time)
- Waktu penghapusan (delete time)
- Penggunaan memori
- Stabilitas kinerja

## HASIL DAN PEMBAHASAN

### A. Analisis Hasil Pengujian

Berdasarkan hasil pengujian, Hash Table menunjukkan waktu pencarian yang lebih cepat dibandingkan Binary Tree untuk pencarian data spesifik.

Hal tersebut terjadi karena Hash Table menggunakan fungsi hash yang secara langsung mengarahkan data ke lokasi penyimpanan.

Sementara itu, Binary Tree memerlukan proses traversal berdasarkan struktur pohon sehingga kompleksitas rata-ratanya mencapai:

**\*O(log n)\***

### B. Perbandingan Kinerja

Operasi Tree	Hash Table	Binary
Pencarian	<b>0(1)</b>	<b>0(log n)</b>
Penyisipan	<b>0(1)</b>	<b>0(log n)</b>
Penghapusan	<b>0(1)</b>	<b>0(log n)</b>

Hash Table memiliki performa tinggi pada pencarian data spesifik, sedangkan Binary Tree memiliki keunggulan pada kestabilan kinerja dan pengelolaan data terurut.

### C. Analisis Penggunaan Memori

Hash Table memerlukan alokasi array tambahan untuk penyimpanan indeks hasil hashing.

Binary Tree memerlukan memori tambahan untuk menyimpan pointer pada setiap node.

### Pembahasan

Hash Table lebih cocok digunakan pada:

- Sistem login
- Database pelanggan
- Sistem cache
- Pencarian berdasarkan ID

Binary Tree lebih cocok digunakan pada:

- Sistem basis data
- Pengurutan data
- Pencarian rentang nilai
- Analisis data besar

## **KESIMPULAN**

Berdasarkan hasil penelitian, Hash Table memiliki keunggulan pada kecepatan pencarian data spesifik dengan kompleksitas rata-rata  $O(1)$ . Namun performanya dapat menurun apabila terjadi collision dalam jumlah besar.

Binary Tree memiliki keunggulan dalam pengelolaan data terurut dan kestabilan pencarian dengan kompleksitas  $O(\log n)$  pada kondisi pohon seimbang.

Pemilihan struktur data yang digunakan harus disesuaikan dengan kebutuhan sistem dan karakteristik data yang diproses.

## **Saran**

Penelitian selanjutnya dapat menggunakan jumlah data yang lebih besar dan menambahkan perbandingan dengan struktur data lain seperti B-Tree, Trie, atau Graph.

## **DAFTAR PUSTAKA**

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.
- Dilber, U. (n.d.). Hash Tables vs Binary Search Trees: The Ultimate Search Battle. Medium. [<https://medium.com/@ugurdilber1/hash-tables-vs-binary-search-trees-the-ultimate-search-battle-74e20a92c206>](<https://medium.com/@ugurdilber1/hash-tables-vs-binary-search-trees-the-ultimate-search-battle-74e20a92c206>)
- Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). *Data Structures and Algorithms in Python*. Wiley.
- Weiss, M. A. (2013). *Data Structures and Algorithm Analysis in C++* (4th ed.). Pearson Education.