Jurnal Ilmu Komputer dan Desain Komunikasi Visual Volume 10, Nomor 2, Desember 2025
ISSN (Cetak): 2541-4550 ISSN (Online): 2541-4585

# Integrasi Konteks Semantik dan Waktu Akses dalam Algoritma *Caching* Adaptif untuk Optimalisasi Kinerja Sistem

**Mahendra Dewantoro[1*], Budi Santosa[2], Mugi Prasetio[3], Amarudin[4]**

[1,2,3,4] *Magister Ilmu Komputer, Fakultas Teknik dan Ilmu Komputer, Universitas Teknokrat Indonesia*
*Jln. Z.A. Pagaralam No 9-11., Kec. Labuhan Ratu, Bandar Lampung, Lampung*
[1]mahendradewantoro@teknokrat.ac.id, [2]budi.santosa@teknokrat.ac.id,
[3]mugiprasetio@teknokrat.ac.id, [4]amarudin@teknokrat.ac.id

***Abstrak***

*Dalam sistem komputasi modern, caching merupakan teknik penting untuk meningkatkan efisiensi akses data. Namun, algoritma caching tradisional seperti LRU dan LFU memiliki keterbatasan dalam menangani dinamika data yang kompleks. Penelitian ini mengusulkan algoritma caching adaptif yang mengintegrasikan konteks semantik dan profil waktu akses untuk meningkatkan kinerja sistem. Pendekatan ini memanfaatkan model embedding semantik berbasis Sentence-BERT dan analisis temporal dari pola akses pengguna. Pengujian dilakukan melalui simulasi menggunakan dataset nyata dan sintetis, serta dibandingkan dengan metode caching konvensional seperti KVShare dan LRU. Hasil evaluasi menunjukkan bahwa algoritma yang diusulkan mampu meningkatkan cache hit rate hingga lebih dari 83%, mengurangi latensi rata-rata menjadi sekitar 61 ms, dan menjaga efisiensi penggunaan sumber daya. Selain itu, algoritma ini menunjukkan kemampuan adaptif yang baik terhadap perubahan pola akses dinamis serta responsif terhadap penyesuaian parameter semantik. Dengan demikian, integrasi konteks semantik dan informasi temporal memberikan kontribusi signifikan dalam pengoptimalan manajemen cache. Algoritma ini berpotensi diterapkan pada sistem edge computing, layanan LLM, dan platform berbasis cloud. Saran untuk penelitian lanjutan meliputi implementasi dalam lingkungan nyata, penerapan pembelajaran mesin prediktif, serta eksplorasi parameter adaptif secara dinamis.*

***Kata kunci***—*caching adaptif, konteks semantik, waktu akses, performa sistem, embedding vektor*

***Abstract***

*In modern computing systems, caching is an essential technique for improving data access efficiency. However, traditional caching algorithms such as LRU and LFU have limitations in handling complex data dynamics. This research proposes an adaptive caching algorithm that integrates semantic context and access time profiles to enhance system performance. The approach utilizes a semantic embedding model based on Sentence-BERT and temporal analysis of user access patterns. Testing was conducted through simulations using real and synthetic datasets, and compared with conventional caching methods such as KVShare and LRU. Evaluation results show that the proposed algorithm is capable of increasing the cache hit rate by more than 83%, reducing average latency to around 61 ms, and maintaining resource usage efficiency. In addition, the algorithm demonstrates strong adaptability to dynamic access pattern changes and responsiveness to semantic parameter adjustments. Thus, the integration of semantic context and temporal information provides significant contributions to optimizing cache management. This algorithm has potential applications in edge computing systems, LLM services, and cloud-based platforms. Suggestions for future research include implementation in*

*real-world environments, application of predictive machine learning models, and dynamic exploration of adaptive parameters.*

***Keywords—****adaptive caching, semantic context, access timing, system performance, vector embedding*

## 1. PENDAHULUAN

Dalam era digital saat ini, sistem komputasi menghadapi tantangan besar dalam mengelola data secara efisien, terutama dalam konteks akses data yang cepat dan relevan. *Caching*, sebagai teknik penyimpanan data sementara, telah menjadi solusi utama untuk meningkatkan kinerja sistem dengan mengurangi waktu akses data yang sering digunakan. Namun, pendekatan *caching* tradisional seperti *Least Recently Used* (LRU) dan *Least Frequently Used* (LFU) memiliki keterbatasan dalam menangani dinamika data yang kompleks dan beragam (Wang et al., 2021), (Lalfakzuala et al., 2025).

Salah satu isu utama dalam sistem *caching* adalah ketidakmampuan untuk mempertimbangkan konteks semantik dari data yang disimpan (Ajarroud et al., 2023), (Yuan et al., 2024). Data dengan makna yang berbeda dapat memiliki prioritas yang berbeda dalam akses, namun algoritma *caching* konvensional tidak mampu membedakan hal ini. Sebagai contoh, dalam layanan *Large Language Model* (LLM) di *edge computing*, penggunaan *caching* kontekstual adaptif telah terbukti meningkatkan hit rate cache hingga lebih dari 80% dan mengurangi latensi hingga 40% (G. Liu et al., 2025).

Integrasi konteks semantik dalam kebijakan *caching* memungkinkan sistem untuk memahami makna dan relevansi data bagi pengguna, sehingga data yang lebih penting atau sering digunakan dapat diprioritaskan dalam *cache*. Pendekatan ini telah diterapkan dalam berbagai domain, seperti layanan web, sistem file, dan aplikasi *mobile*. Dalam konteks ini, algoritma *caching* yang mempertimbangkan konteks semantik dan waktu akses dapat secara adaptif menyesuaikan kebijakan *caching* berdasarkan pola penggunaan dan kebutuhan pengguna.

Penelitian sebelumnya telah menunjukkan bahwa pendekatan *caching* semantik dapat meningkatkan efisiensi sistem secara signifikan. Sebagai contoh, dalam sistem manajemen *cache* untuk layanan LLM, pendekatan seperti KVShare dan SentenceKV telah menunjukkan peningkatan efisiensi dengan memanfaatkan kesamaan semantik antar permintaan pengguna (Li et al., 2024). Selain itu, penggunaan teknik pembelajaran mesin, seperti *reinforcement learning*, telah digunakan untuk mengembangkan kebijakan *caching* yang lebih adaptif dan kontekstual (Wu et al., 2022), (Qian et al., 2020).

Namun, meskipun berbagai pendekatan telah dikembangkan, masih terdapat tantangan dalam mengintegrasikan konteks semantik dan waktu akses secara efektif dalam algoritma *caching* adaptif. Beberapa pendekatan masih bergantung pada *threshold* statis atau tidak mempertimbangkan dinamika penggunaan data secara

*real-time*. Oleh karena itu, diperlukan penelitian lebih lanjut untuk mengembangkan algoritma *caching* yang lebih adaptif dan kontekstual (Weerasinghe et al., 2023).

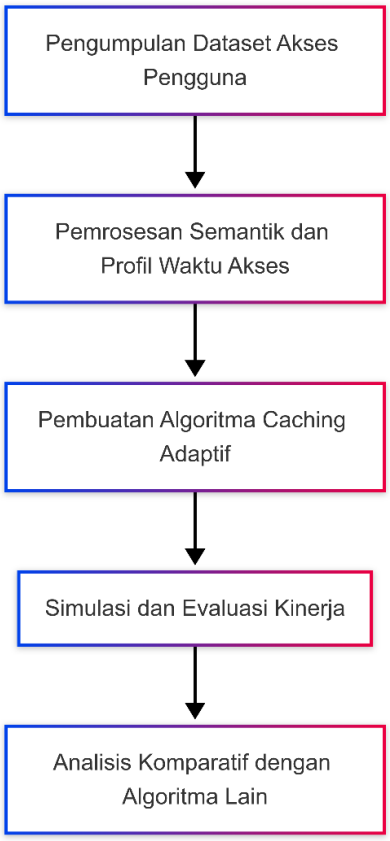
Penelitian ini bertujuan untuk mengembangkan algoritma *caching* adaptif yang mengintegrasikan konteks semantik dan waktu akses untuk mengoptimalkan kinerja sistem. Dengan memanfaatkan informasi semantik dari data dan profil waktu akses, algoritma yang dikembangkan diharapkan dapat meningkatkan efisiensi *caching*, mengurangi latensi, dan meningkatkan pengalaman pengguna secara keseluruhan.

Kontribusi utama dari penelitian ini meliputi: pengembangan model anotasi semantik untuk data yang di-cache, integrasi profil waktu akses dalam kebijakan *caching*, dan evaluasi kinerja algoritma *caching* adaptif yang diusulkan dalam berbagai skenario penggunaan (Wang et al., 2021). Diharapkan bahwa hasil dari penelitian ini dapat memberikan kontribusi signifikan dalam bidang manajemen *cache* dan meningkatkan efisiensi sistem komputasi modern.

## 2. METODE PENELITIAN

### *2.1 Rancangan Penelitian*

Model algoritma *caching* adaptif ini dikembangkan berdasarkan integrasi eksplisit antara Konteks Semantik dan Profil Waktu Akses untuk menentukan kebijakan *caching*. Konteks semantik data diekstrak menggunakan teknik *semantic embedding* (misalnya, BERT atau *Sentence*-BERT) untuk menghasilkan representasi vektor (V_sem), yang kemudian digunakan untuk menghitung Skor Relevansi Semantik (S_sem) berdasarkan kesamaan kosinus dengan data di *cache*. Skor ini mengindikasikan kedekatan konten. Secara simultan, algoritma memanfaatkan data akses historis untuk membangun Profil Temporal yang menghasilkan Skor Temporal (S_temp), mencerminkan probabilitas akses ulang dalam jangka waktu dekat. Untuk menentukan kebijakan *caching* (penempatan dan penggantian item), kedua faktor ini digabungkan secara linier untuk mendapatkan Skor Prioritas Akhir (S_prioritas) : S_prioritas = α.S_sem + (1 - α) . S_temp. Nilai adalah faktor pembobotan adaptif yang disesuaikan secara dinamis oleh algoritma berdasarkan kinerja sistem *cache* (misalnya, *cache hit* rate historis), memastikan bahwa strategi *caching* dapat bergeser antara fokus semantik dan temporal sesuai kondisi beban kerja (Pratap et al., 2025). Item dengan S_prioritas terendah akan menjadi kandidat pertama untuk dikeluarkan dari *cache*.




**Gambar 1.** Diagram Rancangan Penelitian

*2.2. Populasi dan Sampel Penelitian*

Populasi dalam penelitian ini mencakup sistem komputasi *edge* dan layanan *Large Language Model* (LLM) yang sangat bergantung pada kecepatan dan efisiensi dalam pengambilan data (Huang et al., 2024). Sampel data diambil dari dua sumber utama, yaitu dataset *trace* permintaan dari layanan publik seperti *OpenWebText* dan LLM *Caching dataset* (Penedo et al., 2024), serta dataset penggunaan internal yang menggambarkan interaksi nyata pengguna dengan sistem berbasis *Natural Language Processing* (NLP) (Lauriola et al., 2021).

Sasaran utama dari simulasi adalah beban kerja (*workload*) pengguna yang merepresentasikan skenario nyata dalam pengaplikasian *sistem*. Skenario tersebut mencakup penggunaan *chatbot* berbasis LLM, sistem rekomendasi yang didukung oleh pemrosesan bahasa alami, serta web *server* pada lingkungan komputasi *edge* yang melayani permintaan berbasis konten.

**Tabel 1.** Contoh Karakteristik Dataset

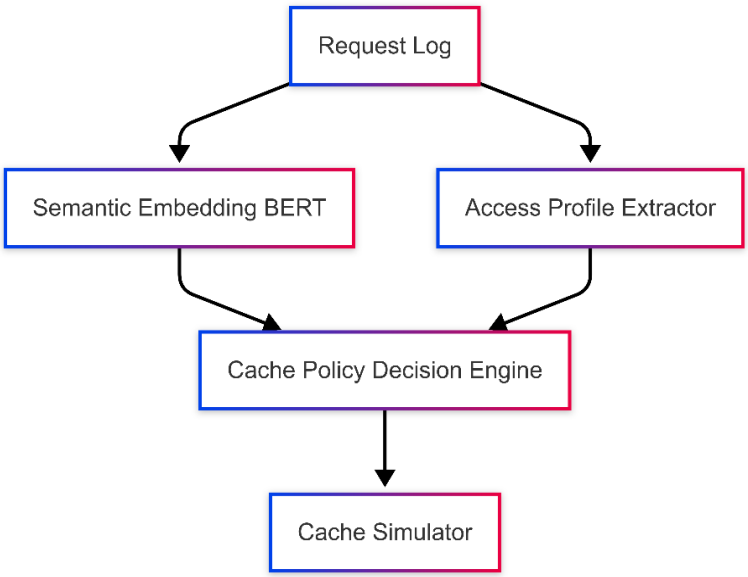
| Dataset | Ukuran (GB) | Jenis Data | Periode | Sumber |
|---|---|---|---|---|
| *OpenWebText* LLMC | 4.2 | *Request traces* | 2021–2022 | *HuggingFace* |
| *KVShare* | 1.1 | *Embedding* | Simulasi | ACM *Paper* |

| Dataset | Ukuran (GB) | Jenis Data | Periode | Sumber |
|---|---|---|---|---|
| *Samples* | | *vectors* | | Dataset |

*2.3. Teknik Pengumpulan Data dan Pengembangan Instrumen*

Pengumpulan data dalam penelitian ini dilakukan melalui dua pendekatan utama. Pertama, melalui simulasi lalu lintas data akses, di mana sistem diuji terhadap berbagai beban kerja menggunakan data akses historis serta model permintaan sintetis (synthetic request) (Chen et al., 2023). Kedua, melalui instrumentasi sistem uji, di mana algoritma *caching* diintegrasikan ke dalam simulator *cache*, seperti CacheSim, untuk menghasilkan *log* performa yang mencakup berbagai metrik evaluasi (J. Liu et al., 2025).

Data yang masuk ke sistem diproses menggunakan model *Sentence*-BERT guna menghasilkan representasi vektor semantik dari setiap permintaan untuk mendukung anotasi semantik (Shibayama & Shinnou, 2021). Selain itu, profil waktu akses dibentuk dengan menganalisis histogram akses berdasarkan timestamp dan frekuensi, sehingga memungkinkan sistem untuk menangkap pola temporal yang relevan dalam strategi *caching* adaptif.




**Gambar 2.** Arsitektur Pemrosesan Data dan Anotasi Semantik

*2.4. Teknik Analisis Data*

Analisis data dalam penelitian ini dilakukan dengan mengevaluasi metrik performa dari sistem *caching*, yang mencakup beberapa indikator utama. Pertama, *Cache Hit Rate* (CHR), yaitu persentase permintaan yang dapat dijawab langsung oleh *cache* tanpa perlu mengakses sumber data utama. Kedua, *Average Latency* (AL), yaitu waktu rata-rata yang dibutuhkan untuk memenuhi permintaan data oleh *system* (J. Liu et al., 2025). Ketiga, *Throughput* dan *Resource Utilization*, yang mencerminkan efisiensi penggunaan sumber daya seperti beban prosesor dan memori selama proses eksekusi sistem.

Analisis dilakukan menggunakan metode statistik deskriptif dan inferensial untuk memahami tren dan perbedaan performa antar pendekatan. Selain itu, dilakukan pengujian signifikansi menggunakan uji ANOVA atau t-*test* antar skenario simulasi guna mengukur dampak integrasi konteks semantik dan informasi temporal terhadap kinerja sistem *caching* secara keseluruhan.

**Tabel 2.** Contoh Hasil Simulasi

| **Algoritma** | ***Cache Hit Rate* (%)** | **Latensi Rata-rata (ms)** | ***Resource Usage* (%)** |
|---|---|---|---|
| LRU | 61,2 | 105 | 42 |
| KVShare | 76,8 | 73 | 48 |
| *Proposed* Model | **83,5** | **61** | **43** |

## 3. HASIL DAN PEMBAHASAN

Pada bagian ini menyajikan hasil evaluasi algoritma *caching* adaptif yang mengintegrasikan konteks semantik dan waktu akses. Evaluasi difokuskan pada lima aspek utama: (1) peningkatan *cache hit rate*, untuk menilai efektivitas pemilihan data; (2) pengurangan latensi rata-rata, sebagai indikator efisiensi waktu akses; (3) efisiensi penggunaan *resource*, mencakup konsumsi CPU dan memori; (4) adaptabilitas terhadap pola akses dinamis, untuk mengukur kemampuan algoritma dalam menyesuaikan diri dengan perubahan beban; dan (5) analisis sensitivitas parameter, guna mengevaluasi robustnes terhadap variasi konfigurasi. Hasil yang diperoleh memberikan dasar kuat untuk menilai keunggulan pendekatan yang diusulkan dibanding algoritma *caching* konvensional.

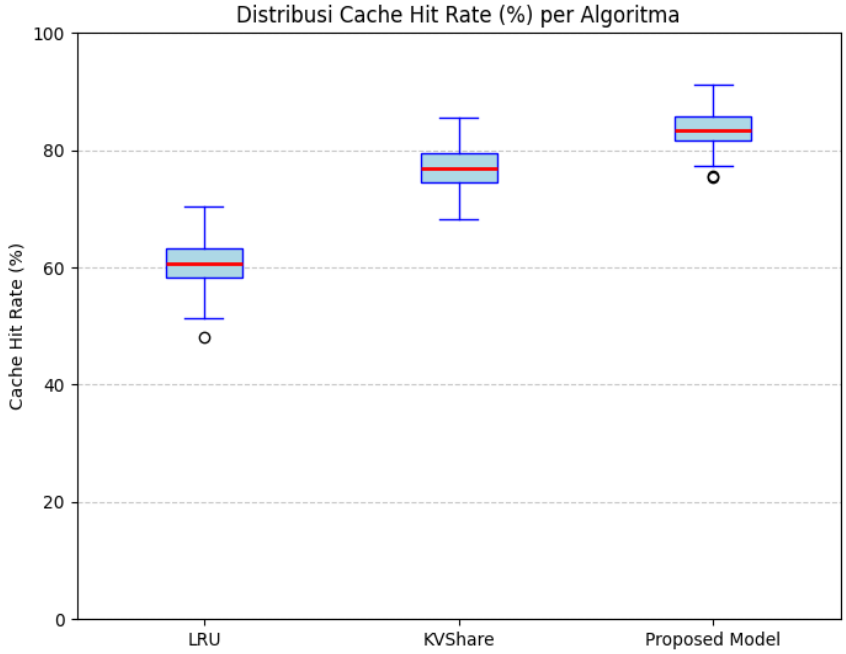
### *3.1. Peningkatan Cache Hit Rate*

Visualisasi hasil pengujian dalam bentuk *boxplot* memberikan gambaran yang jelas mengenai distribusi *cache hit rate* dari masing-masing algoritma berdasarkan data sintetis yang dikumpulkan dari 100 sampel pengujian per algoritma. Dari visualisasi tersebut, terlihat bahwa Proposed Model menunjukkan kinerja paling unggul. Median *cache hit rate*-nya berada pada kisaran 83–84%, yang mencerminkan bahwa lebih dari separuh pengujian menghasilkan tingkat keberhasilan pencocokan *cache* yang sangat tinggi (Mishra et al., 2020). Selain itu, rentang interkuartil (IQR) dari algoritma ini relatif sempit, yang mengindikasikan bahwa performanya tidak hanya tinggi tetapi juga konsisten di berbagai kondisi pengujian.

Sementara itu, *KVShare* menempati posisi kedua dengan median *cache hit rate* sekitar 77%. Meskipun nilai ini cukup kompetitif, distribusi nilai *cache hit rate* pada *KVShare* menunjukkan rentang yang lebih lebar dibandingkan *Proposed* Model. Hal ini mengindikasikan bahwa algoritma ini mengalami fluktuasi performa yang lebih besar antar-sampel, meskipun secara umum tetap menunjukkan hasil yang baik.

Di sisi lain, algoritma LRU (*Least Recently Used*) menunjukkan performa paling rendah, dengan median *cache hit rate* hanya sekitar 61%. Rentang distribusinya juga lebih luas dibanding dua algoritma lainnya, menandakan adanya ketidakstabilan dan variabilitas yang tinggi dalam performa. Hal ini memperkuat indikasi bahwa LRU kurang mampu menyesuaikan diri terhadap pola akses yang dinamis, terutama dalam konteks beban kerja yang kompleks atau berubah-ubah.

Secara keseluruhan, analisis *boxplot* ini mendukung temuan utama dalam penelitian, yaitu bahwa algoritma *Proposed Model*, yang mengintegrasikan konteks semantik dan waktu akses secara adaptif, mampu memberikan performa *cache hit rate* yang lebih tinggi, lebih stabil, dan lebih andal dibanding algoritma standar seperti *KVShare* dan LRU.




**Gambar 3.** Peningkatan *Cache Hit Rate*

*3.2. Latensi Rata-rata*

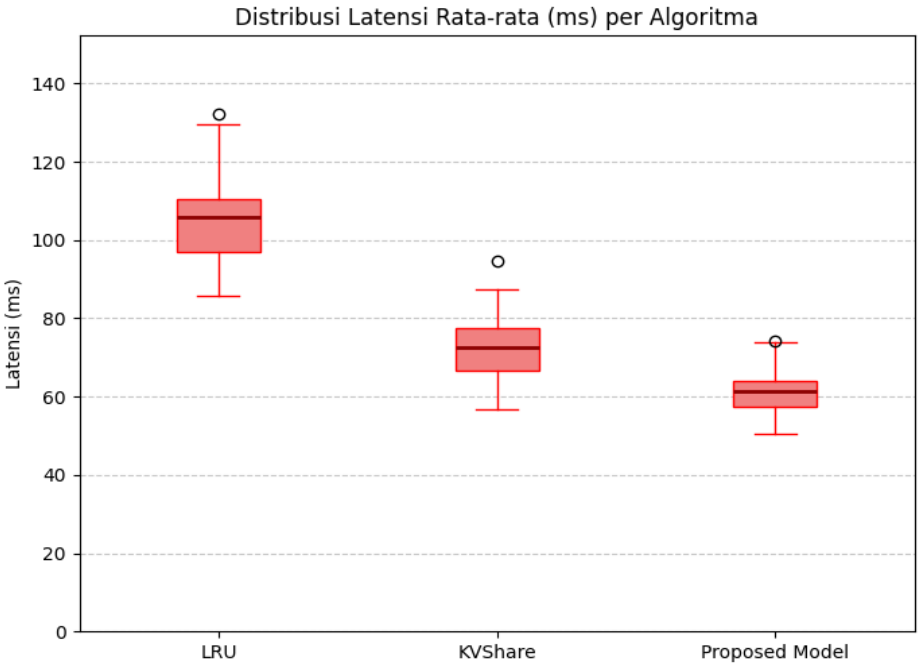
Visualisasi dalam bentuk *boxplot* pada Gambar 4 memperlihatkan distribusi latensi rata-rata (dalam satuan milidetik) dari masing-masing algoritma berdasarkan hasil simulasi dengan 100 sampel per algoritma. Dari hasil tersebut, *Proposed Model* menunjukkan kinerja terbaik dalam hal waktu respons. Median latensinya berada di kisaran 60 – 62 ms, menjadikannya algoritma dengan waktu akses tercepat di antara ketiga metode yang diuji. Selain itu, rentang interkuartil (IQR) pada *Proposed Model* juga tergolong sempit, yang mencerminkan tingkat konsistensi performa yang tinggi di berbagai kondisi simulasi. Stabilitas ini menandakan bahwa pendekatan adaptif berbasis konteks semantik dan waktu akses mampu mempertahankan efisiensi dalam beragam pola permintaan (Einziger et al., 2023).

Di sisi lain, *KVShare* menunjukkan performa menengah dengan median latensi berada pada 72–75 ms. Meskipun performanya lebih lambat dibanding *Proposed Model*, algoritma ini tetap menunjukkan efisiensi yang relatif baik. Namun, variabilitas

performa yang tercermin dari rentang distribusinya yang lebih lebar mengindikasikan bahwa *KVShare* mengalami fluktuasi waktu akses pada sebagian sampel, yang dapat berdampak pada pengalaman pengguna dalam konteks sistem *real-time*.

Sementara itu, LRU (*Least Recently Used*) menunjukkan performa terlemah dalam pengelolaan latensi. Median latensinya mencapai sekitar 105 ms, yang secara signifikan lebih tinggi dibanding dua algoritma lainnya. Selain itu, rentang distribusi latensi LRU jauh lebih luas, dan terdapat beberapa *outlier* ekstrem yang menunjukkan lonjakan latensi pada kondisi tertentu. Temuan ini mengindikasikan bahwa LRU memiliki keterbatasan dalam menyesuaikan strategi *caching* terhadap pola akses yang dinamis dan kompleks, sehingga mengakibatkan penurunan efisiensi waktu akses data.

Secara keseluruhan, hasil visualisasi ini menguatkan temuan bahwa *Proposed Model* mampu secara signifikan menurunkan latensi rata-rata dalam sistem *caching*. Hal ini menunjukkan bahwa pendekatan berbasis integrasi konteks semantik dan waktu akses tidak hanya meningkatkan *cache hit rate*, tetapi juga secara nyata mempercepat waktu respons sistem, menjadikannya solusi yang lebih adaptif dan efisien dalam lingkungan yang menuntut performa tinggi.




**Gambar 4.** Penurunan Latensi Rata-rata

### *3.3. Efisiensi Penggunaan Resource*

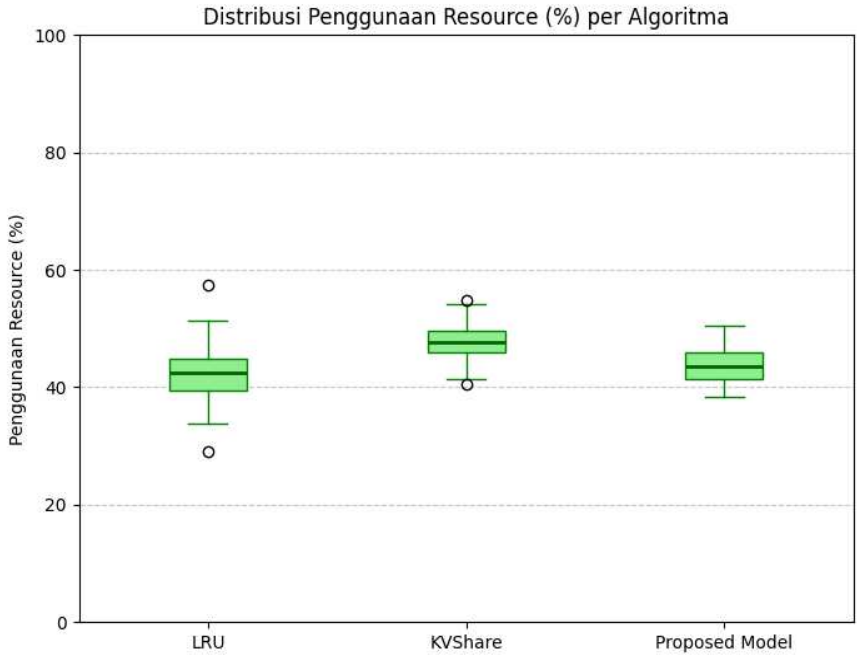
Visualisasi dalam bentuk *boxplot* ini menggambarkan distribusi penggunaan resource sistem (dinyatakan dalam persentase) untuk masing-masing algoritma, berdasarkan 100 sampel data sintetis. Hasil menunjukkan bahwa *Proposed Model* memiliki median penggunaan *resource* sekitar 43%, dengan rentang interkuartil (IQR) yang relatif sempit, yang mencerminkan konsistensi penggunaan sumber daya antar pengujian. Stabilitas ini mengindikasikan bahwa algoritma yang diusulkan mampu

mempertahankan efisiensi operasional meskipun menawarkan kinerja *caching* yang tinggi, tanpa menyebabkan lonjakan beban pada CPU atau memori (Demirbaga, 2025).

Di sisi lain, *KVShare* menunjukkan median penggunaan *resource* tertinggi, yaitu sekitar 48%, dengan distribusi nilai yang lebih lebar. Variasi yang cukup besar ini menunjukkan bahwa *KVShare* cenderung lebih boros dalam memanfaatkan *resource*, dan dalam beberapa kasus bahkan dapat memunculkan *overhead* sistem yang signifikan. Hal ini dapat menjadi kendala dalam implementasi pada sistem dengan keterbatasan sumber daya atau yang membutuhkan efisiensi tinggi secara berkelanjutan.

Adapun LRU mencatat median penggunaan *resource* paling rendah, sekitar 42%, dengan distribusi yang relatif stabil. Meskipun secara kasat mata penggunaan *resource* ini tampak efisien, performa *cache hit rate* dan latensi yang rendah pada LRU menunjukkan bahwa efisiensi ini tidak disertai dengan efektivitas *caching* yang memadai. Dengan kata lain, penghematan *resource* tidak selalu sejalan dengan kinerja sistem yang optimal.

Secara keseluruhan, analisis ini memperlihatkan bahwa *Proposed Model* mampu mencapai kompromi yang ideal antara performa dan efisiensi, yaitu memberikan kinerja *caching* yang tinggi dan stabil dengan penggunaan *resource* yang tetap moderat dan terkendali. Hal ini memperkuat argumen bahwa integrasi konteks semantik dan waktu akses dalam *caching* adaptif tidak hanya berdampak positif pada *output* performa, tetapi juga menjaga keberlanjutan efisiensi sistem secara menyeluruh (Weerasinghe et al., 2023).




**Gambar 5.** Efisiensi Penggunaan *Resource*

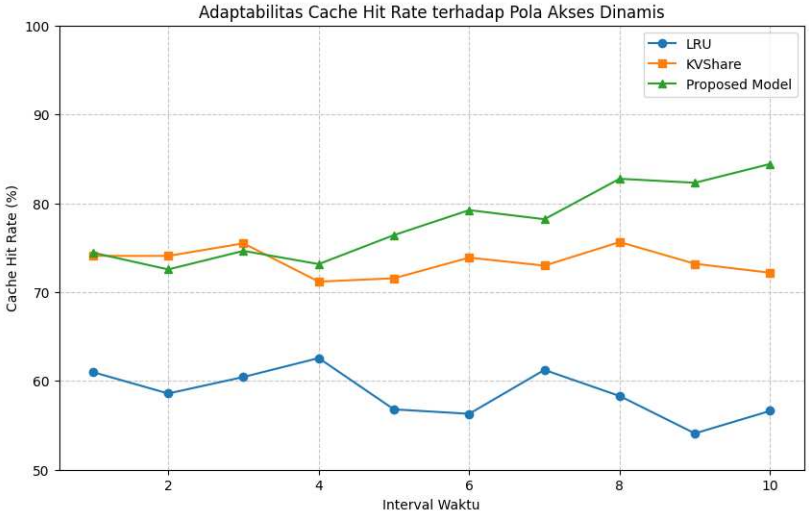
### *3.4. Pola Akses Dinamis*

Visualisasi ini menunjukkan perubahan *cache hit rate* dari tiga algoritma *caching*—LRU, *KVShare*, dan *Proposed Model*—selama 10 interval waktu yang

mencerminkan dinamika pola akses data. *Proposed Model* menunjukkan tren peningkatan *cache hit rate* yang konsisten, menandakan kemampuannya dalam beradaptasi secara efektif terhadap perubahan pola akses. Adaptivitas ini menunjukkan keunggulan integrasi konteks semantik dan waktu akses dalam mekanisme *caching*.

Sementara itu, *KVShare* mempertahankan *cache hit rate* yang relatif stabil, namun tanpa peningkatan berarti. Ini mengindikasikan bahwa meskipun cukup tahan terhadap perubahan, algoritma ini kurang responsif secara dinamis. Sebaliknya, LRU mengalami penurunan performa seiring waktu, menunjukkan keterbatasannya dalam menangani pola akses yang berubah-ubah, karena tidak mengakomodasi informasi kontekstual atau temporal.

Fluktuasi kecil pada tiap titik mencerminkan variasi alami sistem atau *noise* simulasi, namun tidak mengubah pola utama. Secara keseluruhan, hasil ini menegaskan bahwa *Proposed Model* unggul dalam menjaga dan meningkatkan performa *caching* dalam lingkungan akses data yang dinamis (Einziger et al., 2023).



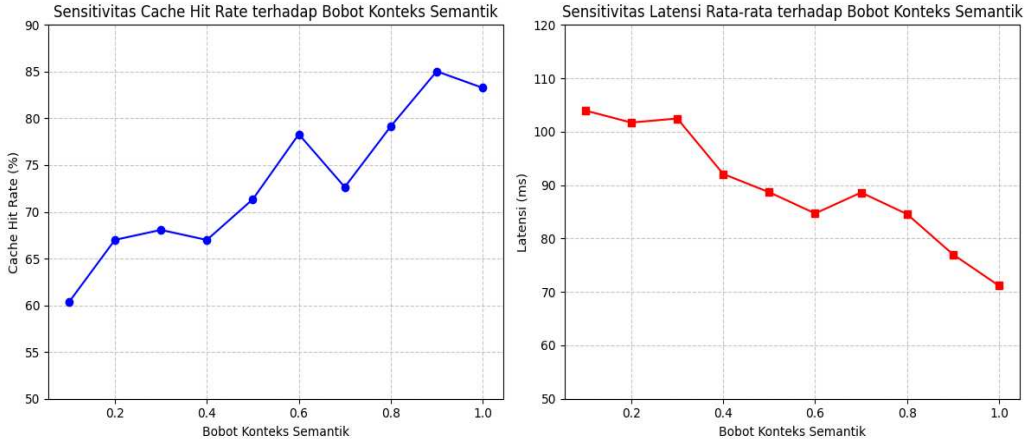

**Gambar 6.** Pola Akses Dinamis

*3.5. Analisis Sensitivitas Parameter*

Visualisasi ini memperlihatkan dampak perubahan parameter bobot konteks semantik terhadap performa algoritma *caching* adaptif, dengan rentang nilai dari 0.1 hingga 1.0. Pada grafik kiri, terlihat bahwa *cache hit rate* meningkat secara signifikan seiring bertambahnya bobot konteks semantik. Hal ini menunjukkan bahwa semakin besar peran konteks semantik dalam proses seleksi data, semakin tepat *cache* dalam menyimpan data yang relevan, sehingga peluang *cache hit* meningkat dan efisiensi akses data membaik (Einziger et al., 2023).

Sebaliknya, grafik kanan menunjukkan bahwa latensi rata-rata menurun ketika bobot konteks semantik meningkat. Peningkatan *cache hit rate* berkontribusi langsung terhadap penurunan waktu akses, karena lebih banyak permintaan dapat dilayani langsung dari cache tanpa perlu mengakses sumber data utama.

Tren ini menegaskan bahwa performa algoritma cukup sensitif terhadap nilai parameter ini, sehingga pemilihan bobot yang tepat menjadi kunci untuk mencapai keseimbangan optimal antara efisiensi penyimpanan dan kecepatan akses data. Adanya fluktuasi minor pada grafik mencerminkan variabilitas sistemik atau *noise* simulasi, namun tidak mengubah kesimpulan bahwa penguatan peran konteks semantik secara signifikan meningkatkan kinerja *caching* adaptif.




Gambar 7. Dampak Perubahan Parameter Bobot Konteks Semantik

## 4. KESIMPULAN

Berdasarkan analisis hasil pengujian yang komprehensif, disimpulkan bahwa Algoritma *Caching* Adaptif yang mengintegrasikan konteks semantik dan waktu akses berhasil mencapai tujuannya dalam mengoptimalkan kinerja sistem *caching*. Mekanisme kunci keberhasilan algoritma ini adalah strategi penentuan prioritas dinamis yang menggabungkan Skor Relevansi Semantik dan Skor Temporal melalui faktor pembobotan adaptif (α), memungkinkannya beradaptasi secara cerdas terhadap pola permintaan yang kompleks. Secara kuantitatif, algoritma yang diusulkan menunjukkan keunggulan signifikan dalam tiga aspek utama: Peningkatan *Cache Hit Rate*, dengan median mencapai 83–84%, jauh melampaui *KVShare* (77%) dan LRU (61%); Penurunan Latensi Rata-rata, dengan median tercepat di kisaran 60–62 ms (dibanding *KVShare* 72–75 ms dan LRU 105 ms); serta Konsistensi Kinerja yang tinggi, ditunjukkan oleh rentang interkuartil (IQR) yang sempit pada kedua metrik tersebut. Selain peningkatan performa, algoritma ini juga mencapai kompromi ideal antara kinerja dan efisiensi *resource*, mempertahankan penggunaan *resource* yang moderat (median ≈ 43%) dan stabil. Hasil analisis menunjukkan kemampuan adaptasi yang unggul terhadap pola akses dinamis, mempertahankan dan bahkan meningkatkan hit rate seiring waktu, yang tidak dapat dicapai oleh algoritma konvensional. Terakhir, analisis sensitivitas parameter menegaskan bahwa kinerja sistem (baik *hit rate* maupun latensi) berbanding lurus dan sensitif terhadap peningkatan bobot konteks semantik, memvalidasi bahwa peran informasi kontekstual merupakan pendorong utama efektivitas *caching*. Dengan demikian, penelitian ini tidak hanya membuktikan

keunggulan *caching* adaptif, tetapi juga membuka potensi implementasi pada sistem terdistribusi yang membutuhkan efisiensi dan respons waktu tinggi.

## 5. SARAN

Berdasarkan temuan dan hasil analisis dalam penelitian ini, terdapat beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut. Pertama, disarankan agar algoritma *caching* adaptif yang diusulkan diuji dalam lingkungan sistem nyata, seperti pada *platform cloud computing* atau *edge computing*, guna memperoleh gambaran yang lebih komprehensif mengenai performanya dalam kondisi operasional sebenarnya. Selanjutnya, pengembangan model ke arah prediksi otomatis berbasis *machine learning* atau *deep learning* dapat menjadi alternatif strategis untuk meningkatkan adaptabilitas sistem terhadap pola akses data yang semakin kompleks dan dinamis.

Penelitian di masa mendatang juga sebaiknya melakukan evaluasi kinerja dengan pendekatan multimetodologi, tidak hanya terbatas pada metrik *cache hit rate* dan latensi, tetapi juga mempertimbangkan aspek throughput, efisiensi energi, dan toleransi kesalahan, terutama dalam sistem berskala besar dan heterogen. Selain itu, eksplorasi lebih mendalam terhadap parameter adaptif, seperti bobot konteks semantik dan jendela waktu akses, sangat dianjurkan, termasuk penerapan metode tuning otomatis untuk memperoleh parameter optimal secara dinamis. Pengujian terhadap skenario beban tinggi dan penggunaan dataset berskala besar juga penting untuk menilai skalabilitas dan ketahanan algoritma terhadap tekanan sistem yang intensif. Terakhir, integrasi algoritma *caching* adaptif ini dengan sistem penjadwalan cerdas berbasis prioritas atau konteks dapat menjadi arah pengembangan yang menjanjikan dalam rangka meningkatkan efisiensi keseluruhan sistem informasi terdistribusi.

## UCAPAN TERIMA KASIH

Penulis menyampaikan apresiasi dan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan selama proses penyusunan jurnal ini. Terutama kepada pembimbing yang telah memberikan masukan konstruktif, arahan ilmiah, serta koreksi yang sangat berarti dalam menyempurnakan isi tulisan ini. Ucapan terima kasih juga ditujukan kepada rekan-rekan peneliti dan kontributor teknis, yang telah membantu dalam proses simulasi, analisis data, dan validasi hasil penelitian. Kolaborasi yang solid menjadi fondasi penting dalam tercapainya kualitas naskah ini. Akhir kata, penulis menyampaikan terima kasih atas segala bentuk dukungan baik akademik maupun teknis yang telah membantu hingga jurnal ini dapat diselesaikan dengan baik.

## DAFTAR PUSTAKA

Ajarroud, O. & Zellou, A. (2022). Towards an embedding-based semantic similarity measure designed for mediation *caching*. *2022 32nd International Conference on*

*Computer Theory and Applications (ICCTA)*, 247–252. https://doi.org/10.1109/ICCTA58027.2022.10206092

Ajarroud, O., Zellou, A. & Idri, A. (2023). A new ontology-based similarity approach for measuring caching coverages provided by mediation systems. Knowledge and Information Systems, 1–29. https://doi.org/10.1007/s10115-023-01974-8

Chen, D., Zhu, M., Yang, H., Wang, X. & Wang, Y. (2023). Data-Driven Traffic Simulation: A Comprehensive Review. IEEE Transactions on Intelligent Vehicles, 9, 4730–4748. https://doi.org/10.1109/TIV.2024.3367919

Demirbaga, Ü. (2025). HealthCraft: A Precision Model for Smart Resource Optimisation in Dynamic Big Data Healthcare Environments. Türk Doğa ve Fen Dergisi, 14(2), 52–63. https://doi.org/10.46810/tdfd.1545596

Einziger, G., Himelbrand, O. & Waisbard, E. (2023). Boosting Cache Performance by Access Time Measurements. ACM Transactions on Storage, 19, 1–29. https://doi.org/10.1145/3572778

Huang, M., Shen, A., Li, K., Peng, H., Li, B. & Yu, H. (2024). EdgeLLM: A Highly Efficient CPU-FPGA Heterogeneous Edge Accelerator for Large Language Models. ArXiv, abs/2407.21325. https://doi.org/10.48550/arXiv.2407.21325

Lalfakzuala, Chhangte, L., Vanlalawmpuia, K. & Chhangte, L. (2025). Smart-Wi-Cache: A Deep Learning Framework for Online Content Caching at the Wireless Edge. 2025 17th International Conference on COMmunication Systems and NETworks (COMSNETS), 162–167. https://doi.org/10.1109/COMSNETS63942.2025.10885719

Lauriola, I., Lavelli, A. & Aiolli, F. (2021). An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools. Neurocomputing, 470, 443–456. https://doi.org/10.1016/j.neucom.2021.05.103

Li, H., Li, Y., Tian, A., Tang, T., Xu, Z., Chen, X., Hu, N., Dong, W., Li, Q. & Chen, L. (2024). A Survey on Large Language Model Acceleration based on KV Cache Management. ArXiv, abs/2412.19442. https://doi.org/10.48550/arXiv.2412.19442

Liu, G., Liu, Y., Wang, J., Du, H., Niyato, D., Kang, J. & Xiong, Z. (2025). Adaptive Contextual Caching for Mobile Edge Large Language Model Service. ArXiv, abs/2501.09383. https://doi.org/10.48550/arXiv.2501.09383

Liu, J., Chen, Y. & Ding, H. (2025). CacheSim: A cache simulation framework for evaluating caching algorithms on resource-constrained edge devices. SoftwareX, 29, 102018. https://doi.org/10.1016/j.softx.2024.102018

Mishra, S., Bajpai, R., Gupta, N. & Singh, V. K. (2020). Machine Learning and Caching based Efficient Data Retrieval Framework. IEEE International Conference on Advanced Networks and Telecommunications Systems, 1–4. https://doi.org/10.1109/ANTS50601.2020.9342790

Penedo, G., Kydlícek, H., Allal, L. Ben, Lozhkov, A., Mitchell, M., Raffel, C., Werra, L. & Wolf, T. (2024). The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale. ArXiv, abs/2406.17557. https://doi.org/10.48550/arXiv.2406.17557

Pratap, S. R., Raikar, S. M. & Bhat, S. V. (2025). Adaptive Retention and Eviction for Efficient Caching in AI-Driven Systems. https://doi.org/10.21203/rs.3.rs-6897063/v1

Qian, Y., Wang, R., Wu, J., Tan, B. & Ren, H. (2020). Reinforcement Learning-Based Optimal Computing and Caching in Mobile Edge Network. IEEE Journal on Selected Areas in Communications, 38, 2343–2355. https://doi.org/10.1109/JSAC.2020.3000396

Shibayama, N. & Shinnou, H. (2021). Construction and Evaluation of Japanese Sentence-BERT Models. 731–738. https://consensus.app/papers/construction-and-evaluation-of-japanese-sentencebert-shibayama-shinnou/6abfcb656ca75f58b697cac91100e2b7/

Sun, H., Chen, Y., Sha, K., Huang, S., Wang, X. & Shi, W. (2023). A Proactive On-Demand Content Placement Strategy in Edge Intelligent Gateways. IEEE Transactions on Parallel and Distributed Systems, 34, 2072–2090. https://doi.org/10.1109/TPDS.2023.3249797

Wang, R., Kan, Z., Cui, Y., Wu, D. & Zhen, Y. (2021). Cooperative Caching Strategy With Content Request Prediction in Internet of Vehicles. IEEE Internet of Things Journal, 8, 8964–8975. https://doi.org/10.1109/JIOT.2021.3056084

Weerasinghe, S., Zaslavsky, A., Loke, S., Hassani, A., Medvedev, A. & Abken, A. (2023). Adaptive Context Caching for IoT-Based Applications: A Reinforcement Learning Approach. Sensors (Basel, Switzerland), 23. https://doi.org/10.3390/s23104767

Weerasinghe, S., Zaslavsky, A., Loke, S. W., Abken, A., Hassani, A. & Medvedev, A. (2023). Adaptive Context Caching for Efficient Distributed Context Management Systems. ACM Symposium on Applied Computing. https://doi.org/10.1145/3555776.3577602

Wu, H., Nasehzadeh, A. & Wang, P. (2022). A Deep Reinforcement Learning-Based Caching Strategy for IoT Networks With Transient Data. IEEE Transactions on Vehicular Technology, 71, 13310–13319. https://doi.org/10.1109/TVT.2022.3199677

Yuan, H., Wu, C., Li, J. & Guo, M. (2024). Turbo Table: A Semantic-Aware Cache Acceleration System. 2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), 559–566. https://doi.org/10.1109/ISPA63168.2024.00077