



Tinjauan Literatur terhadap Manajemen Sumber Daya pada Sistem Operasi yang Menggunakan Arsitektur Microkernel

Muhammad Adil Keysa Andra¹, Abdul Majid²

^{1,2}Universitas Islam Negeri K.H. Abdurrahman Wahid Pekalongan

E-mail: muhammad.adil.keysa.andra24032@mhs.uingusdur.ac.id¹,
abdulmajid1183@gmail.com²

Article Info

Article history:

Received December 12, 2025

Revised December 14, 2025

Accepted December 19, 2025

Keywords:

Microkernel, Resource
Management, IPC, CPU
Scheduling, Memory
Management

ABSTRACT

The complexity of modern operating systems has led to an increase in the size and functionality of kernels in monolithic architectures, which has resulted in an increased attack surface and the risk of system-wide failure. As an alternative, microkernel architectures embrace the principle of minimalism by limiting kernel functions to only essential services, while other components are run in user space. This article aims to review and analyze resource management—particularly CPU scheduling and memory management—in operating systems that use microkernel architecture. The method used is qualitative research with a narrative literature review approach to relevant scientific journals, conference proceedings, and technical documentation. The results of the study show that microkernels offer significant advantages in terms of error isolation, security, and maintainability. However, this architecture also faces challenges in the form of performance overhead due to the intensity of inter-process communication (IPC). With modern optimizations such as fast-path IPC and zero-copy, microkernels remain relevant for systems that demand high reliability, such as embedded and real-time systems.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Article Info

Article history:

Received December 12, 2025

Revised December 14, 2025

Accepted December 19, 2025

Kata Kunci:

Microkernel, Manajemen
Sumber Daya, IPC,
Penjadwalan CPU, Manajemen
Memori

ABSTRACT

Kompleksitas sistem operasi modern mendorong peningkatan ukuran dan fungsi kernel pada arsitektur monolitik, yang berimplikasi pada meningkatnya permukaan kesalahan (attack surface) serta risiko kegagalan sistem secara menyeluruh. Sebagai alternatif, arsitektur microkernel mengusung prinsip minimalisme dengan membatasi fungsi kernel hanya pada layanan esensial, sementara komponen lain dijalankan di ruang pengguna. Artikel ini bertujuan untuk meninjau dan menganalisis manajemen sumber daya—khususnya penjadwalan CPU dan manajemen memori—pada sistem operasi yang menggunakan arsitektur microkernel. Metode yang digunakan adalah penelitian kualitatif dengan pendekatan *narrative literature review* terhadap jurnal ilmiah, prosiding konferensi, dan dokumentasi teknis yang relevan. Hasil kajian menunjukkan bahwa microkernel menawarkan keunggulan signifikan dalam hal isolasi kesalahan, keamanan, dan *maintainability*. Namun demikian, arsitektur ini juga menghadapi tantangan berupa *overhead* kinerja akibat intensitas komunikasi antar-proses (*Inter-Process Communication/IPC*). Dengan adanya optimasi modern seperti *fast-path IPC* dan *zero-copy*, microkernel tetap relevan untuk sistem yang menuntut keandalan tinggi, seperti sistem *embedded* dan *real-time*.



This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Muhammad Adil Keysa Andra

Universitas Islam Negeri K.H. Abdurrahman Wahid Pekalongan

Email: muhammad.adil.keysa.andra24032@mhs.uingusdur.ac.id

PENDAHULUAN

Sistem operasi (SO) memiliki peran fundamental sebagai pengelola sumber daya utama dalam sistem komputer, termasuk CPU, memori, dan perangkat input/output. Seiring meningkatnya kompleksitas perangkat keras dan kebutuhan aplikasi modern, desain sistem operasi turut berkembang untuk menjawab tuntutan kinerja, keamanan, dan keandalan. Sebagian besar sistem operasi populer dikembangkan menggunakan arsitektur monolitik, di mana berbagai layanan inti seperti manajemen berkas, driver perangkat, dan subsistem jaringan berjalan dalam satu ruang alamat kernel. Pendekatan ini menawarkan kinerja tinggi, tetapi memiliki risiko besar karena kesalahan pada satu komponen dapat menyebabkan kegagalan sistem secara keseluruhan (Tanenbaum & Bos, 2015).

Sebagai alternatif, arsitektur microkernel diperkenalkan dengan filosofi meminimalkan fungsi yang berjalan di dalam kernel. Hanya mekanisme inti seperti penjadwalan dasar, manajemen memori tingkat rendah, dan komunikasi antar-proses yang dijalankan di kernel, sementara layanan lainnya ditempatkan di ruang pengguna sebagai proses terpisah. Pendekatan ini bertujuan meningkatkan modularitas, keamanan, dan isolasi kesalahan. Konsep microkernel telah dikaji sejak pengembangan Mach dan L4, serta terus disempurnakan melalui implementasi modern seperti seL4, MINIX 3, dan Zircon yang menunjukkan penerapan praktis dalam sistem nyata (Heiser & Elphinstone, 2016).

Permasalahan dan Signifikansi

Pemindahan layanan sistem ke ruang pengguna pada arsitektur microkernel mengubah paradigma manajemen sumber daya dari terpusat menjadi terdistribusi. Perubahan ini menimbulkan sejumlah permasalahan, antara lain perbedaan mekanisme penjadwalan CPU dibandingkan arsitektur monolitik, dampak overhead komunikasi antar-proses terhadap kinerja sistem, serta kemampuan microkernel dalam menjamin ketersediaan sumber daya saat terjadi kegagalan komponen. Analisis terhadap permasalahan ini penting untuk menentukan relevansi microkernel dalam berbagai domain komputasi, khususnya pada sistem yang menuntut keandalan dan keamanan tinggi.

Tujuan Artikel

Penelitian ini bertujuan untuk: (1) meninjau mekanisme manajemen sumber daya pada arsitektur microkernel, (2) menganalisis dampak IPC terhadap kinerja sistem, dan (3) mengkaji keunggulan microkernel dalam aspek isolasi kesalahan dan *fault tolerance* berdasarkan kajian literatur yang ada.

METODE PENELITIAN

Penelitian ini menggunakan metode kualitatif dengan pendekatan studi kepustakaan (*literature review*) yang bersifat deskriptif dan komparatif. Jenis *literature review* yang



digunakan adalah *narrative literature review*, yaitu metode peninjauan pustaka yang bertujuan menyajikan pemahaman komprehensif terhadap suatu topik melalui analisis dan sintesis hasil penelitian terdahulu tanpa prosedur meta-analisis statistik.

Sumber data penelitian berupa data sekunder yang diperoleh dari artikel jurnal ilmiah, prosiding konferensi internasional, buku teks utama bidang sistem operasi, serta dokumentasi teknis resmi pengembang microkernel. Literatur dipilih berdasarkan relevansi dengan topik manajemen sumber daya, kredibilitas sumber, dan kontribusinya terhadap pemahaman arsitektur microkernel.

Analisis data dilakukan secara deskriptif-kualitatif dengan mengidentifikasi konsep utama, mengelompokkan temuan berdasarkan tema, membandingkan hasil penelitian antar sumber, serta mensintesis kesimpulan mengenai efektivitas dan tantangan manajemen sumber daya pada sistem operasi berarsitektur microkernel.

HASIL DAN PEMBAHASAN

Desain Microkernel: Konsep Dasar dan Implikasi

Microkernel dirancang dengan memisahkan mekanisme dan kebijakan (*mechanism and policy separation*). Kernel hanya menyediakan mekanisme dasar, sementara kebijakan pengelolaan sumber daya diimplementasikan di tingkat server ruang pengguna. Pendekatan ini meningkatkan fleksibilitas dan modularitas sistem, tetapi juga menambah kebutuhan komunikasi antar-komponen melalui IPC.

Penjadwalan CPU dan Overhead IPC

Dalam arsitektur microkernel, penjadwalan CPU dapat diimplementasikan secara lebih fleksibel melalui server penjadwalan di ruang pengguna. Namun, intensitas IPC dan *context switching* berpotensi meningkatkan latensi dan konsumsi siklus CPU. Penelitian awal pada microkernel generasi pertama menunjukkan penurunan kinerja dibandingkan kernel monolitik, tetapi implementasi modern seperti L4 dan seL4 telah berhasil menekan *overhead* tersebut melalui optimasi *fast-path IPC* dan *zero-copy* (Liedtke, 1995).

Manajemen Memori dan Fault Tolerance

Microkernel memanfaatkan mekanisme isolasi memori untuk memastikan bahwa setiap komponen sistem berjalan dalam ruang alamat terpisah. Dengan demikian, kegagalan pada satu driver atau layanan tidak langsung memengaruhi kernel maupun komponen lain. Pendekatan ini memungkinkan pemulihan komponen secara dinamis dan meningkatkan *fault tolerance*, sebagaimana ditunjukkan dalam desain MINIX 3 yang mendukung *self-healing* sistem (Tanenbaum & Bos, 2015).

KESIMPULAN

Berdasarkan hasil tinjauan literatur, dapat disimpulkan bahwa arsitektur microkernel menawarkan paradigma manajemen sumber daya yang lebih aman dan modular dibandingkan arsitektur monolitik. Meskipun terdapat *trade-off* berupa *overhead* kinerja akibat IPC, perkembangan optimasi microkernel modern telah mengurangi dampak tersebut secara signifikan. Oleh karena itu, microkernel tetap relevan untuk sistem yang membutuhkan keandalan, keamanan, dan isolasi kesalahan yang tinggi, khususnya pada domain *embedded* dan *real-time*.



DAFTAR PUSTAKA

- Heiser, G., & Elphinstone, K. (2016). L4 microkernels: The lessons from 20 years of research and deployment. *ACM Transactions on Computer Systems*, 34(1).
- Liedtke, J. (1995). On micro-kernel construction. In *Proceedings of the 15th ACM Symposium on Operating Systems Principles*.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley.
- Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson.