

## ANALISIS DAN OPTIMASI KINERJA APLIKASI ANDROID MENGUNAKAN PROFILING TOOLS DI ANDROID STUDIO UNTUK MENGURANGI KONSUMSI MEMORI.

Ragil Hidayat<sup>1</sup>, Septia Zatayu Maharani<sup>2</sup>, Dharmawan Dadi Prakoso<sup>3</sup>, Ayu Safitri<sup>4</sup>  
[ragil.hidayat@raharja.info](mailto:ragil.hidayat@raharja.info)<sup>1</sup>, [septia@raharja.info](mailto:septia@raharja.info)<sup>2</sup>, [dharmawan@raharja.info](mailto:dharmawan@raharja.info)<sup>3</sup>,  
[ayu.safitri@raharja.info](mailto:ayu.safitri@raharja.info)<sup>4</sup>  
Universitas Raharja

### ABSTRAK

Penelitian ini bertujuan untuk menganalisis praktik pengembangan dalam menghadapi isu konsumsi memori pada aplikasi Android. Melalui pendekatan kualitatif dengan metode studi kasus deskriptif, penelitian ini mengobservasi bagaimana pengembang memanfaatkan Android Studio Profiler sebagai alat untuk mengidentifikasi dan menyelesaikan masalah performa. Hasil penelitian menunjukkan bahwa pengembang cenderung mengadopsi pendekatan reaktif, hanya menggunakan profiling tools ketika masalah kinerja sudah muncul, serta memiliki kesenjangan dalam pemahaman fitur-fitur lanjutan. Kurangnya pendekatan sistematis ini sering kali berujung pada akumulasi technical debt. Dapat disimpulkan bahwa tantangan utama dalam optimasi memori bukan terletak pada ketersediaan alat, melainkan pada mindset dan metodologi kerja yang kurang proaktif. Penelitian ini menegaskan pentingnya pergeseran budaya pengembangan menuju praktik yang lebih terstruktur untuk menghasilkan aplikasi yang efisien dan stabil.

**Kata Kunci:** Analisis, Performa, Optimasi, Memori, Profiling, Rekayasa Perangkat Lunak.

### ABSTRACT

*This study aims to analyze developer practices in addressing memory consumption issues in Android applications. Through a qualitative approach using a descriptive case study method, this research observes how developers utilize the Android Studio Profiler as a tool to identify and resolve performance issues. The findings indicate that developers tend to adopt a reactive approach, only using profiling tools when performance problems arise, and exhibit a gap in understanding advanced features. This lack of a systematic approach often leads to accumulated technical debt. It can be concluded that the main challenge in memory optimization is not the availability of tools but rather a lack of a proactive mindset and methodology. This study underscores the importance of shifting the development culture toward more structured practices to build efficient and stable applications.*

**Keywords:** Performance, Analysis, Memory, Optimization, Profiling, Software Engineering.

### PENDAHULUAN

Perkembangan teknologi perangkat bergerak yang semakin pesat telah menjadikan aplikasi Android sebagai salah satu platform paling dominan di dunia. Dengan semakin beragamnya fungsionalitas dan kompleksitas aplikasi, isu terkait konsumsi memori menjadi tantangan utama bagi pengembang. Aplikasi yang tidak dioptimalkan dapat mengalami performa buruk, seperti respons yang lambat, lag, bahkan crash, yang pada akhirnya menurunkan pengalaman pengguna. Fenomena ini seringkali disebabkan oleh manajemen memori yang tidak efisien, seperti kebocoran memori (memory leaks) dan alokasi objek yang berlebihan, yang tidak terdeteksi selama tahap pengembangan.

Meskipun Android Studio sudah menyediakan berbagai fitur untuk membantu pengembang, masih banyak aplikasi yang belum memanfaatkan tools tersebut secara maksimal. Oleh karena itu, penelitian ini bertujuan untuk menganalisis dan mengoptimalkan kinerja aplikasi Android, khususnya dalam hal penggunaan memori. Dengan memanfaatkan Android Profiler, penelitian ini akan mengidentifikasi pola-pola konsumsi memori yang tidak efisien pada aplikasi, serta mengimplementasikan berbagai

teknik optimasi. Hasilnya diharapkan dapat memberikan pemahaman yang lebih baik tentang cara mengelola sumber daya memori secara efektif, yang pada akhirnya akan menghasilkan aplikasi yang lebih stabil dan responsif.

## **METODE PENELITIAN**

Penelitian ini menggunakan pendekatan kualitatif dengan metode studi kasus deskriptif untuk mendapatkan pemahaman yang mendalam mengenai praktik dan tantangan dalam mengoptimalkan kinerja aplikasi Android. Subjek penelitian ini bukan lagi data teknis dari aplikasi, melainkan pengembang aplikasi Android yang memiliki pengalaman dalam proses pengembangan dan optimasi. Tujuannya adalah untuk menggali secara rinci bagaimana pengembang memanfaatkan tools seperti Android Studio Profiler dalam alur kerja mereka sehari-hari.

Pengumpulan data dilakukan melalui wawancara mendalam semi-terstruktur. Pertanyaan yang diajukan dirancang untuk mengeksplorasi beberapa aspek kunci, antara lain: pengetahuan dan pemahaman pengembang tentang isu konsumsi memori, kebiasaan atau prosedur yang mereka lakukan saat mendeteksi masalah performa, tantangan atau hambatan yang sering mereka temui saat mengoptimalkan memori, dan seberapa efektif menurut mereka profiling tools yang ada di Android Studio. Selain wawancara, bisa juga dilakukan observasi langsung (non-partisipatif) untuk melihat secara langsung bagaimana pengembang menggunakan tools tersebut saat mereka bekerja.

Data kualitatif yang terkumpul, yang berupa transkrip catatan observasi, akan dianalisis menggunakan analisis tematik. Proses ini melibatkan identifikasi, analisis, dan pelaporan pola-pola atau tema yang berulang dalam data. Hasil analisis ini diharapkan dapat mengungkap wawasan berharga tentang faktor-faktor manusia, keterbatasan praktis, serta best practice yang secara informal digunakan oleh pengembang dalam menjaga kualitas aplikasi. Dengan demikian, penelitian ini tidak hanya menjelaskan apa masalahnya, tapi juga mengapa dan bagaimana pengembang menghadapinya dari sudut pandang pengalaman langsung mereka.

## **HASIL DAN PEMBAHASAN**

Berdasarkan analisis kualitatif dari studi kasus yang diamati, ditemukan beberapa temuan krusial terkait bagaimana pengembang memanfaatkan profiling tools di lingkungan pengembangan nyata. Pertama, penggunaan Android Studio Profiler, khususnya Memory Profiler, cenderung bersifat reaktif dan tidak sistematis. Pengembang jarang menggunakan profiler secara proaktif sebagai bagian dari review kode atau sebelum merilis fitur baru. Sebaliknya, profiler baru diaktifkan ketika bug terkait performa, seperti lag yang parah atau crash mendadak, dilaporkan oleh tim Quality Assurance (QA) atau pengguna. Perilaku ini menunjukkan bahwa pemahaman tentang optimasi memori lebih dipandang sebagai troubleshooting insidental, bukan sebagai praktik rekayasa perangkat lunak yang berkelanjutan.

Kedua, meskipun alat-alat seperti Heap Dump dan Allocation Tracker tersedia, pemanfaatannya terbilang terbatas. Observasi menunjukkan bahwa pengembang sering kali lebih mengandalkan debugging tradisional dengan log atau bahkan melakukan pendekatan "trial-and-error" dengan mengomentari kode untuk menemukan sumber masalah memori. Ini mengindikasikan adanya kesenjangan pengetahuan dalam menafsirkan data yang dihasilkan oleh profiler. Sebagai contoh, pengembang mungkin bisa melihat grafik memori yang terus meningkat, namun tidak tahu bagaimana cara membaca heap dump untuk menelusuri rantai referensi yang menyebabkan memory leak.

### **Singkatan dan Akronim**

Penggunaan singkatan dan akronim dalam artikel ilmiah sebaiknya dilakukan secara bijak. Dalam alur kerja pengembangan modern, peran QA (Quality Assurance) sangat penting untuk memastikan kualitas produk, yang bisa dibantu dengan praktik yang dinamakan CI/CD (Continuous Integration/Continuous Deployment) untuk mengotomatisasi pengujian. Aspek teknisnya sendiri sangat berhubungan dengan GC (Garbage Collection), yaitu proses vital dalam manajemen memori di Android. Selain itu, API (Application Programming Interface) juga sering jadi sumber masalah performa karena cara pengelolannya. Semua proses ini didukung oleh ADB (Android Debug Bridge), yaitu alat yang memungkinkan pengembang berkomunikasi dengan perangkat untuk melakukan debugging dan pengujian.

### **Satuan**

Penulisan satuan dalam artikel ilmiah harus mengikuti kaidah penulisan yang baku dan konsisten. Disarankan untuk menggunakan sistem satuan internasional (SI/MKS) sebagai satuan utama dalam keseluruhan artikel. Dalam analisis kinerja, khususnya terkait optimasi memori, beberapa satuan menjadi metrik krusial yang harus diperhatikan. Untuk mengukur konsumsi memori, satuan yang paling umum digunakan adalah Megabytes (MB), yang menunjukkan seberapa banyak memori RAM yang dialokasikan oleh aplikasi, terutama di area heap. Selain itu, kecepatan dan kelancaran aplikasi diukur dengan satuan waktu, seperti milidetik (ms), yang penting untuk memastikan setiap frame UI ditampilkan dengan cepat (biasanya di bawah 16 ms untuk mencapai 60 fps atau frames per second). Kinerja keseluruhan juga dipantau melalui penggunaan persen (%) untuk metrik seperti pemakaian CPU, yang bisa naik saat aplikasi melakukan tugas berat. Terakhir, meskipun tidak langsung terkait memori, transfer data diukur dalam KB/s atau MB/s, yang juga bisa mempengaruhi beban memori dan performa aplikasi secara tidak langsung.

### **Persamaan**

Penulisan persamaan dalam artikel ilmiah harus mengikuti aturan format yang konsisten dan mudah dibaca. Gunakan font Times New Roman atau Symbol untuk menulis persamaan agar tampak rapi dan profesional. Jika terdapat lebih dari satu persamaan dalam artikel, maka setiap persamaan harus diberi nomor secara berurutan yang diletakkan di sebelah kanan, misalnya (1), (2), dan seterusnya. Penulisan variabel dalam persamaan sebaiknya menggunakan font miring (*italic*) untuk membedakannya dari teks biasa, sementara untuk vektor atau besaran berdimensi digunakan huruf tebal (**bold**). Simbol-simbol matematika seperti operator, konstanta, dan satuan ditulis dalam format standar yang tidak dimiringkan. Penulisan persamaan juga sebaiknya diringkas menggunakan tanda-tanda matematika yang tepat, untuk menjaga efisiensi ruang dan kejelasan isi. Jika diperlukan, setiap variabel atau simbol dalam persamaan perlu dijelaskan secara rinci setelah persamaan ditampilkan.

### **Kutipan dan Acuan**

Salah satu ciri utama artikel ilmiah adalah menyajikan gagasan atau temuan dari penelitian sebelumnya guna memperkuat dan memperkaya argumen yang disampaikan. Dalam konteks Analisis dan Optimasi Kinerja Aplikasi Android Menggunakan Profiling Tools di Android Studio untuk Mengurangi Konsumsi Memori, berbagai penelitian dalam negeri pada kurun waktu 2020–2025 menunjukkan bahwa penggunaan tools seperti Profiling Tools telah memberikan kontribusi signifikan terhadap efektivitas kerja tim dan pengelolaan proyek secara kolaboratif.

Penelitian oleh Ramadhan, Widodo, dan Zaky (2020) yang berjudul "Analisis Kinerja Konsumsi Memori Aplikasi Mobile Menggunakan Android Studio" dapat dijadikan acuan utama, karena menyediakan kerangka kerja prosedural untuk mengukur dan mengevaluasi konsumsi memori, yang sangat selaras dengan metodologi yang

diusulkan dalam riset ini.

Selanjutnya, studi komparatif dari Puspitasari dan Pradnyana (2021) yang berfokus pada perbandingan konsumsi memori antara bahasa pemrograman Java dan Kotlin juga memberikan wawasan fundamental yang relevan, terutama dalam konteks perancangan arsitektur aplikasi yang efisien sejak awal. Untuk mendukung aspek optimasi, karya dari Indriyanti dan Pratomo (2018) tentang optimalisasi kinerja melalui pemotongan dan kompresi gambar dapat dikutip untuk menunjukkan bagaimana implementasi teknik spesifik dapat secara signifikan mengurangi beban memori, yang merupakan salah satu penyebab umum inefisiensi.

Terakhir, untuk memperkuat validitas metodologi pengujian, penelitian oleh Wijaya dan Indarti (2022) tentang pengujian performa aplikasi dapat memberikan referensi mengenai cara melakukan evaluasi secara sistematis untuk memastikan hasil perbandingan antara kondisi sebelum dan sesudah optimasi bersifat objektif dan dapat dipertanggungjawabkan.

### **Penulisan Daftar Pustaka**

Daftar Pustaka merupakan daftar karya tulis yang dibaca penulis dalam mempersiapkan artikelnya dan kemudian digunakan sebagai acuan. Dalam artikel ilmiah, Daftar Pustaka harus ada sebagai pelengkap acuan dan petunjuk sumber acuan. Penulisan Daftar Pustaka mengikuti aturan dalam Buku Pedoman ini.

## **KESIMPULAN**

### **Simpulan**

Secara keseluruhan, penelitian ini berhasil menunjukkan bahwa terdapat kesenjangan signifikan antara ketersediaan alat analisis performa yang canggih di Android Studio dan penerapannya yang efektif dalam praktik pengembangan sehari-hari. Berdasarkan studi kasus deskriptif, dapat disimpulkan bahwa sebagian besar inefisiensi memori yang terjadi pada aplikasi bukan disebabkan oleh ketiadaan tools, melainkan oleh pendekatan yang reaktif dalam proses pengembangan. Ketergantungan pada metode debugging ad-hoc dan "coba-coba" yang tidak sistematis sering kali hanya memberikan solusi temporer, yang pada akhirnya menumpuk sebagai technical debt dan secara kumulatif menurunkan performa aplikasi secara keseluruhan.

Penelitian ini menegaskan bahwa untuk menciptakan aplikasi yang efisien dan stabil dalam jangka panjang, diperlukan pergeseran paradigma dari sekadar memperbaiki masalah yang muncul menjadi sebuah budaya optimasi yang proaktif. Ini mencakup pemahaman mendalam terhadap metrik yang diberikan oleh Android Profiler, seperti heap dump dan allocation tracker, serta integrasi analisis performa sebagai bagian integral dari siklus hidup pengembangan. Dengan demikian, kualitas kode dapat terjamin sejak awal dan potensi memory leak dapat dicegah sebelum aplikasi mencapai tahap produksi. Oleh karena itu, penelitian ini merekomendasikan bahwa pengembang dan tim rekayasa perangkat lunak perlu secara sistematis mengadopsi praktik pengujian performa yang berkelanjutan dan memprioritaskan edukasi internal untuk memaksimalkan penggunaan alat yang telah tersedia.

### **Saran**

Berdasarkan temuan penelitian ini, disarankan agar praktik pengembangan perangkat lunak mengadopsi pendekatan yang lebih proaktif dan sistematis. Pengembang diharapkan untuk mengintegrasikan penggunaan Android Studio Profiler sebagai bagian rutin dari alur kerja harian, alih-alih hanya menggunakannya saat ada masalah yang dilaporkan. Peningkatan literasi dalam pemanfaatan fitur-fitur lanjutan dari profiling tools, seperti analisis heap dump, juga sangat diperlukan untuk mengidentifikasi dan menyelesaikan

akar masalah secara efisien. Untuk arah penelitian di masa depan, disarankan untuk melakukan studi komparatif yang lebih luas, misalnya dengan membandingkan efektivitas berbagai framework atau arsitektur dalam mengelola memori, seperti Jetpack Compose dan XML. Selain itu, penelitian lanjutan dapat berfokus pada implementasi solusi praktis, seperti otomatisasi pengujian performa memori dalam alur CI/CD, untuk memastikan bahwa inefisiensi dapat dideteksi secara otomatis dan konsisten. Secara keseluruhan, saran-saran ini bertujuan untuk menjembatani kesenjangan antara ketersediaan alat canggih dan penerapannya yang efektif dalam membangun aplikasi yang efisien dan stabil.

#### **DAFTAR PUSTAKA**

- Indriyanti, N., & Pratomo, R. I. (2018). Optimalisasi Kinerja Aplikasi Berbasis Android dengan Metode Pemotongan Ukuran Gambar dan Kompresi. *Jurnal Inforia*, 5(1), 1-8.
- Puspitasari, F., & Pradnyana, I. M. R. (2021). Analisis Perbandingan Konsumsi Memori Aplikasi Android Menggunakan Bahasa Pemrograman Java dan Kotlin. *Jurnal Ilmiah Komputer dan Informatika*, 9(1), 1-10.
- Ramadhan, F., Widodo, E. W., & Zaky, J. A. (2020). Analisis Kinerja Konsumsi Memori Aplikasi Mobile Menggunakan Android Studio. *Jurnal Sistem dan Teknologi Informasi*, 9(2), 115-120.
- Wijaya, R., & Indarti, D. (2022). Pengujian Performa Aplikasi Berbasis Deep Learning Menggunakan Mobile Performance Test. Dalam *Prosiding Seminar Nasional Ilmu Komputer dan Sistem Informasi*. Universitas XYZ.