

Perbandingan Performa Kotlin vs Java dalam Pengembangan Android dengan Metode Iterasi *While*

Janssen Japutra Sanjaya^{1)*}, Joko Susilo²⁾

¹⁾²⁾³⁾Fakultas Informatika dan Komunikasi Jurusan Teknik Informatika Institut Bisnis dan Informatika Kwik Kian Gie Jakarta Utara, Indonesia

¹⁾51210003@student.kwikkiangie.ac.id

²⁾joko.susilo@kwikkiangie.ac.id

Article history:

Received xx Des 2024;
Revised xx Des 2024;
Accepted xx Des 2024;
Available online 27 Des 2024

Keywords:

Android
Java
Kotlin
Mobile Development
Performa Aplikasi

Abstrak

Studi ini membandingkan kinerja Java dan Kotlin dalam pengembangan aplikasi Android, dengan fokus khusus pada efisiensi eksekusi, penggunaan CPU, penggunaan memori, dan kompleksitas kode. Pengujian dilakukan pada perangkat Xiaomi Redmi Note 12 dengan CPU Snapdragon 685 dan RAM 8 GB, yang melibatkan pengembangan aplikasi berbasis iterasi menggunakan perulangan *while*. Hasil penelitian menunjukkan bahwa Kotlin memiliki keunggulan dalam efisiensi kinerja, mampu mengeksekusi rata-rata 170.000 iterasi dalam 10-25 detik, sementara Java hanya mencapai 150.000 iterasi. Dalam hal konsumsi memori, Kotlin menggunakan 125 MB, lebih rendah dibandingkan Java yang mencapai 160 MB. Penggunaan CPU oleh Kotlin hanya sebesar 3%, jauh lebih sedikit dibandingkan Java yang mencapai 40%. Selain itu, Kotlin memerlukan 157 baris kode (LOC), lebih ringkas dibandingkan Java yang memerlukan 169 baris. Secara keseluruhan, Kotlin menawarkan keunggulan dalam efisiensi kinerja, pemanfaatan sumber daya, dan kesederhanaan sintaksis, menjadikannya pilihan yang lebih optimal untuk pengembangan aplikasi Android. Efisiensi ini dapat berdampak positif pada pengurangan konsumsi daya baterai, yang merupakan faktor penting dalam pengembangan aplikasi seluler. Meskipun Java dikenal memiliki stabilitas dan ekosistem yang mapan, penelitian ini memberikan panduan bagi pengembang dalam memilih bahasa pemrograman yang sesuai dengan kebutuhan proyek mereka. Pengembang yang mengutamakan efisiensi, penghematan memori, dan pengurangan kompleksitas kode mungkin lebih memilih Kotlin. Namun, jika stabilitas dan kompatibilitas dengan sistem lama menjadi prioritas, Java tetap menjadi pilihan yang relevan. Dengan hasil ini, pengembang memiliki wawasan yang lebih luas dalam menentukan bahasa yang sesuai dengan kebutuhan proyek mereka, baik dari segi efisiensi kinerja, pemanfaatan sumber daya, maupun kemudahan pemeliharaan kode.

I. PENDAHULUAN

Android adalah salah satu sistem operasi yang paling diminati di seluruh dunia, diterapkan pada berbagai perangkat seperti ponsel, televisi, jam tangan pintar, dan bahkan sistem hiburan di mobil. Dengan dukungan yang luas dari beragam merek terkenal, Android memegang pangsa pasar yang jauh lebih besar dibandingkan sistem operasi lainnya. Salah satu keunggulan utama Android adalah fleksibilitasnya serta kelancaran dalam pengembangannya. Pengembang memiliki kebebasan untuk memanfaatkan dua bahasa pemrograman utama, yaitu Java dan Kotlin, untuk mengembangkan aplikasi pada platform ini.[1]

Kedua Java dan Kotlin dapat digunakan untuk pengembangan aplikasi Android. Namun, sering terjadi perdebatan mengenai perbedaan efisiensi dalam penggunaan sumber daya seperti CPU, memori, dan waktu eksekusi di antara kedua bahasa ini oleh para pengembang. Oleh karena itu, penting untuk melakukan pengujian terhadap kinerja kedua bahasa ini agar pengembang dapat memilih bahasa pemrograman yang paling sesuai dengan kebutuhan proyek mereka.[2]

* Corresponding author

Java, sebuah bahasa pemrograman yang telah ada sejak tahun 1995, menonjol dengan ekosistemnya yang luas. Bahasa ini tidak hanya diaplikasikan dalam pengembangan aplikasi Android, namun juga merambah beragam platform lain seperti aplikasi web, desktop, dan server. Keunggulan utamanya terletak pada kemampuannya dalam penulisan kode sekali yang dapat dijalankan di berbagai platform melalui Java Virtual Machine (JVM). Sebagai pilihan dominan pengembang aplikasi Android selama bertahun-tahun, Java diapresiasi atas stabilitas dan dukungan yang kuat.[3]

Pada tahun 2017, Google memperkenalkan Kotlin sebagai bahasa pemrograman resmi untuk pengembangan aplikasi Android. Kotlin dirancang untuk mengatasi beberapa keterbatasan yang ada pada Java, seperti sintaks yang lebih ringkas, pengelolaan kesalahan null yang lebih aman, serta dukungan untuk fitur-fitur modern, termasuk extension functions dan coroutines. Dengan adanya fitur-fitur tersebut, pengembang dapat menulis kode yang lebih bersih, efisien, dan lebih mudah dikelola dibandingkan dengan menggunakan Java. Sejak saat itu, popularitas Kotlin telah meningkat pesat, dan banyak pengembang mulai beralih dari Java ke Kotlin.[7]

Untuk melakukan pengujian ini, akan dibuat sebuah aplikasi sederhana berbasis iterasi while loop menggunakan kedua bahasa pemrograman. Aplikasi ini akan dijalankan dalam waktu yang telah ditentukan, dan hasil pengujian akan dicatat dan dibandingkan. Dengan menggunakan perangkat keras yang sama selama pengujian, diharapkan hasil yang diperoleh dapat mencerminkan perbandingan kinerja yang objektif dan adil. Diharapkan hasil penelitian ini dapat memberikan wawasan yang lebih mendalam kepada para pengembang dalam pemilihan bahasa pemrograman yang efektif dan efisien untuk pengembangan aplikasi Android. Selain itu, penelitian ini bertujuan untuk mengidentifikasi kekuatan dan kelemahan dari setiap bahasa, memungkinkan para pengembang untuk membuat keputusan yang lebih tepat dalam menentukan teknologi yang sesuai untuk proyek yang sedang mereka kerjakan.[5], [6]

Penelitian ini dimaksudkan untuk membandingkan efisiensi Java dan Kotlin dalam mengolah data integer dan string pada aplikasi Android. Dengan membangun aplikasi sederhana yang berbasis pada iterasi while loop, penelitian akan mengukur penggunaan CPU, memori, dan waktu eksekusi kedua bahasa secara obyektif dengan menggunakan perangkat keras yang sama. Harapannya, hasil penelitian ini akan memberikan wawasan bagi pengembang dalam memilih bahasa yang paling efektif dan efisien sesuai dengan kebutuhan proyek, serta menyajikan identifikasi terhadap kekuatan dan kelemahan masing-masing bahasa tersebut.[4]

II. TINJAUAN PUSTAKA

Mobile Development adalah proses pengembangan aplikasi yang dibuat khusus untuk perangkat seluler, seperti ponsel pintar dan tablet. Proses ini melibatkan tahap desain, pengembangan, pengujian, serta peluncuran aplikasi pada platform tertentu seperti Android, iOS, atau platform hybrid.

Pada penelitian Nikodemus Galih Candra Wicaksono et al. [1] Java adalah salah satu bahasa pemrograman yang populer dengan konsep write once, run anywhere (WORA), yang memungkinkan aplikasi berjalan di berbagai platform tanpa perlu modifikasi besar. Sejak tahun 2017, Kotlin diperkenalkan oleh Google sebagai bahasa resmi untuk pengembangan Android. Kotlin menawarkan sintaks yang lebih ringkas, manajemen null-safety, dan kompatibilitas penuh dengan Java. Penelitian ini menjadi penting untuk mengevaluasi kinerja kedua bahasa dalam pengembangan aplikasi Android, terutama dalam hal waktu eksekusi dan jumlah karakter yang diolah.

Pada penelitian Hayataka Nakamura et al. [5] Kotlin, yang kompatibel dengan Java, mengkompilasi kode menjadi bytecode Java yang dieksekusi oleh Java Virtual Machine (JVM). Berdasarkan studi, terdapat variasi dalam performa pemrosesan loop di Kotlin dan Java tergantung pada metode deskripsi yang digunakan, meskipun keduanya memiliki semantik yang serupa. Pengujian membandingkan performa menjadi dua kategori: cepat dan lambat. Meskipun terdapat perbedaan kecil dalam bytecode, analisis kode asli oleh compiler Just-In-Time (JIT) menunjukkan bahwa bytecode yang lebih cepat dioptimalkan oleh JIT, sementara yang lambat tidak, yang berdampak signifikan pada performa.

Pada penelitian Ratul Sikder et al. [13] Sistem operasi mobile yang paling populer, Android OS, menghadapi kerentanannya seiring dengan peningkatan fungsionalitas dan praktik pengembangan aplikasi yang kurang hati-hati. Google telah memperkenalkan berbagai perbaikan melalui pembaruan, penghapusan, dan pembatasan API tingkat sistem untuk pengembang pihak ketiga guna mengurangi risiko keamanan. Dalam makalah ini, dibahas tentang pengembangan aplikasi tingkat sistem Android, isu privasi, serta langkah-langkah yang perlu dipertimbangkan oleh pengembang untuk menjaga privasi dan meminimalkan risiko keamanan. Selain itu, juga dibahas perkembangan historis Android OS dan peran pengguna akhir dalam melindungi privasi.

Pada penelitian Niclas Everlönn et al. [14] Dalam pengembangan perangkat lunak, berbagai faktor perlu dipertimbangkan, seperti kecepatan, responsivitas, optimisasi, serta kapasitas penyimpanan dan penggunaan oleh konsumen akhir, terutama dalam bidang kritis seperti perawatan medis. Dalam penelitian ini, dibahas perbandingan kinerja antara bahasa pemrograman Java dan Kotlin melalui eksperimen dan tinjauan literatur. Hasil penelitian menunjukkan bahwa secara umum, Java lebih unggul dalam hal kinerja, walaupun Kotlin dapat menunjukkan performa yang lebih baik dalam beberapa benchmark spesifik. Namun, perbedaan kinerja ini tidak begitu signifikan, kecuali jika fitur idiomatik Kotlin diimplementasikan, yang dapat menambahkan overhead yang

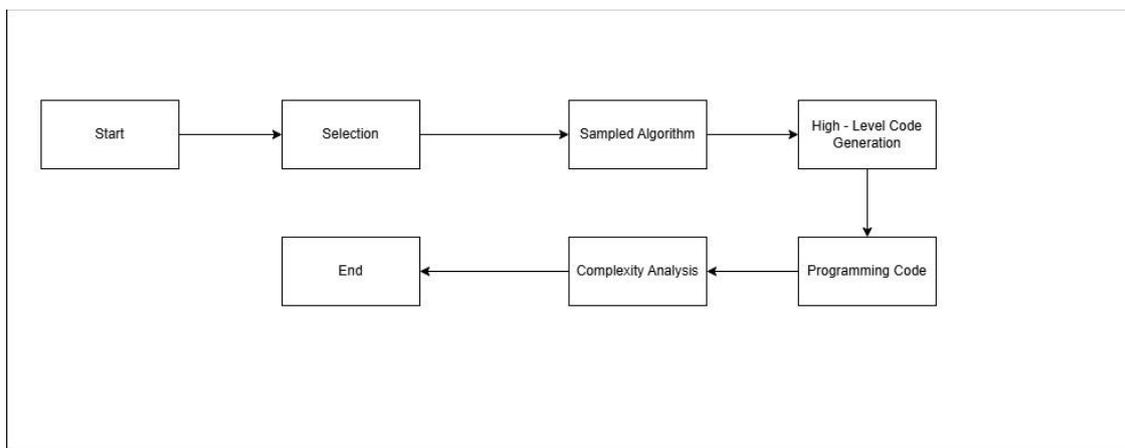
cukup besar. Oleh karena itu, jika kinerja merupakan prioritas utama, penggunaan fitur tersebut harus dilakukan dengan cermat.

Pada penelitian oleh Farihan Elyana Zahari. [15] Kotlin, yang diperkenalkan pada tahun 2019, dan Java, yang telah hadir sejak 1995, keduanya digunakan dalam pengembangan aplikasi Android melalui lingkungan pengembangan resmi, yaitu Android Studio. Sebuah penelitian telah membandingkan kedua bahasa tersebut dari berbagai aspek seperti deklarasi variabel, null safety, kondisi, fungsi, kelas, interoperabilitas, dan lazy loading. Hasil dari penelitian ini menunjukkan bahwa Kotlin lebih sesuai bagi pengembang yang sudah memiliki pengalaman dengan Java karena memiliki fitur modern dan memudahkan dalam penulisan kode. Selain itu, Kotlin juga menawarkan keamanan null yang lebih baik dan kemampuan interoperabilitas yang mempermudah penggunaan bersama dengan kode Java yang sudah ada.

Penelitian lainnya oleh Daniel Sulowski et al. [6] penelitian ini mempertimbangkan perbandingan antara bahasa pemrograman Java dan Kotlin dalam pengembangan aplikasi mobile untuk sistem Android. Analisis meliputi evaluasi kinerja terkait penggunaan CPU, RAM, waktu kompilasi, dan eksekusi. Selain itu, dilibatkan juga faktor-faktor seperti struktur kode, ketersediaan pustaka, dukungan database, popularitas, serta dukungan dari komunitas pengembang.

III. METODE

Metode yang digunakan dalam penelitian ini bertujuan untuk membandingkan kinerja bahasa pemrograman Java dan Kotlin dalam pengolahan tipe data integer dan string pada aplikasi berbasis Android. Proses perbandingan dilakukan melalui serangkaian langkah sistematis yang meliputi iterasi sederhana, pengujian pembuatan string acak, pengukuran waktu eksekusi, serta analisis penggunaan CPU dan memori. Berikut ini adalah penjelasan langkah-langkah metode penelitian yang diilustrasikan pada gambar 1. [7], [4], [12]



Gambar 1 Langkah-langkah penelitian

A. Selections

Langkah awal melibatkan identifikasi operasi inti yang akan diujikan, yang meliputi iterasi pengolahan data integer dan pembuatan string acak. Mekanisme utama yang diterapkan melibatkan fungsi iterasi dan pembuatan string. Performa kedua fungsi ini akan diukur dalam dua bahasa pemrograman, yakni Java dan Kotlin. Fungsi inti seperti iterasi dan pembuatan string diimplementasikan dengan menggunakan perulangan while, dengan parameter waktu eksekusi yang telah ditetapkan (10 dan 25 detik). Proses tersebut akan dijalankan sebanyak 5 kali untuk mendapatkan data yang akurat.

B. Sampled Algorithm

Pada tahap ini, saya mengimplementasikan algoritma iterasi dan pengolahan string menggunakan bahasa Java dan Kotlin. Untuk menguji tipe data integer, saya menggunakan perulangan while dalam algoritma iterasi. Pada setiap iterasi, variabel global iteration akan bertambah hingga mencapai waktu yang ditentukan. Untuk pengolahan string, saya menggunakan metode charAt() dalam Java dan metode random() dalam Kotlin untuk mengacak karakter dari string yang telah disediakan. Saya membandingkan kedua algoritma ini berdasarkan logika, efisiensi, dan kecepatan eksekusinya.

C. High-Level Code Generation

Dalam pengkodean iterasi dan pembuatan string menggunakan Java dan Kotlin, terdapat perbedaan signifikan dalam metode pengacakan string. Pada Java, karakter diacak dengan menggunakan metode charAt(new Random().nextInt(KARAKTER.length())), sedangkan di Kotlin, karakter dipilih secara langsung menggunakan metode random() pada variabel string yang tersedia. Selain itu, perbedaan mendasar juga terletak pada pendekatan

perulangan while, dimana Java menggunakan perulangan while dengan pendekatan yang serupa di Kotlin. Semua kode telah diuji untuk memastikan fungsi berjalan dengan baik dan menghasilkan output yang diinginkan.

D. Programming Corpus

Dalam langkah ini, semua kode dari Java dan Kotlin dikumpulkan dan disusun ke dalam korpus pemrograman. Korpus ini mencakup baris-baris kode yang mencerminkan logika pemrosesan integer dan string. Selanjutnya, dilakukan analisis perbandingan dengan menghitung jumlah baris kode (Lines of Code/LOC) untuk kedua bahasa tersebut. Tujuannya adalah untuk mengidentifikasi perbedaan tingkat kompleksitas pemrograman antara Java dan Kotlin. Kode-kode tersebut juga akan digunakan sebagai acuan dalam analisis lanjutan, termasuk pengukuran efisiensi waktu dan ruang.

E. Complexity Analysis

Dalam Analisis Complexity, proses penyusunan dan pengujian dilakukan terhadap seluruh kode program menggunakan bahasa pemrograman Java dan Kotlin. Pengujian ini dimaksudkan untuk membandingkan performa kedua bahasa dalam memproses data integer dan string. Proses tersebut dilakukan pada perangkat dengan spesifikasi identik guna menjamin obyektivitas hasil pengujian.

IV. HASIL

TABEL 1
 SPESIFIKASI PERANGKAT KERAS

Hardware	Precision
Smartphone	Xiaomi Redmi Note 12
Cpu	Snapdragon 685 Octa-Core (Max 2,80 Ghz)
Ram	8 Gb
Versi Android	Android 14

Dalam tabel 1, pengujian ini melibatkan tiga aspek utama, yakni Kompleksitas Waktu, Pengukuran Kinerja, dan Kompleksitas Kode. Tujuan penelitian ini adalah untuk mengevaluasi dan membandingkan algoritma guna menemukan metode yang paling efisien dalam hal waktu eksekusi, kinerja, dan kesederhanaan kode.

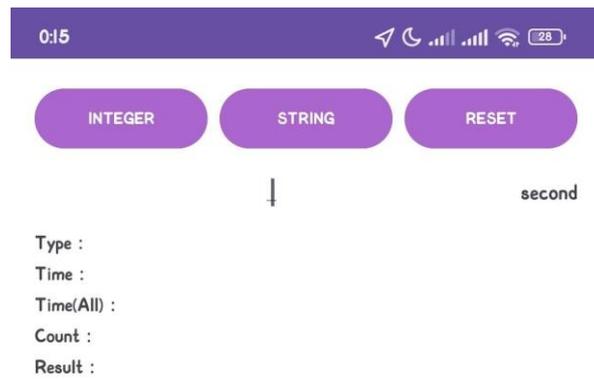
TABEL 2
 PERBANDINGAN KOMPLEKSITAS JAVA DAN KOTLIN

Complexity	Waktu (Nanodetik)	Total (Karakter)
Time Complexity	Eksekusi iterasi: menggunakan while loop.	Eksekusi iterasi: menggunakan while loop
	Pengukuran waktu: System.nanoTime()	Pengukuran waktu: measureNanoTime()
	Durasi pengujian: 10 dan 25 detik.	Durasi pengujian: 10 dan 25 detik.
Space Complexity	Jumlah iterasi rata-rata: 150.000 iterasi.	Jumlah iterasi rata-rata: 170.000 iterasi.
	Penggunaan Memori: 160 MB.	Penggunaan Memori: 125 MB.
	Penggunaan CPU: 40%.	Penggunaan CPU: 3%.
Code Complexity	Lines Of Code (LOC): 169. Membutuhkan inisialisasi komponen tambahan.	Lines Of Code (LOC): 157. Tidak memerlukan inisialisasi komponen.

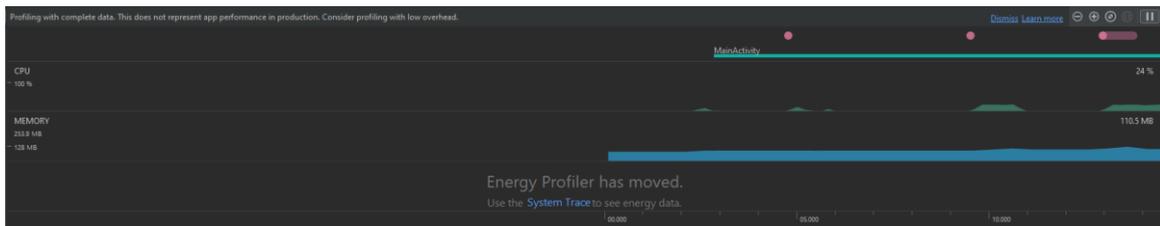
Dalam penelitian ini, dilakukan analisis kompleksitas untuk membandingkan kode pengolahan data integer dan string dengan menggunakan bahasa pemrograman Java dan Kotlin. Perbandingan tersebut mencakup kompleksitas waktu, kompleksitas ruang, dan kompleksitas kode untuk menilai efisiensi masing-masing bahasa pemrograman. Hasil dari analisis kompleksitas kedua bahasa tersebut dapat ditemukan pada Tabel 2.

Dengan melakukan eksperimen langsung pada aplikasi di smartphone Xiaomi Redmi Note 12 dengan CPU Snapdragon 685 Octa-core (Max 2.80GHz), RAM 8 GB, dan versi Android 14, penelitian ini bertujuan untuk mengevaluasi kinerja aplikasi secara praktis di perangkat pengguna. Tujuan utamanya adalah menganalisis efisiensi waktu proses, stabilitas aplikasi, serta keandalan antarmuka pengguna. Melalui eksperimen ini, diharapkan dapat diperoleh wawasan yang mendalam mengenai kecepatan eksekusi, responsivitas aplikasi, dan kompatibilitas dengan perangkat keras yang digunakan. Hasil dari eksperimen ini dapat dilihat pada Gambar 2.

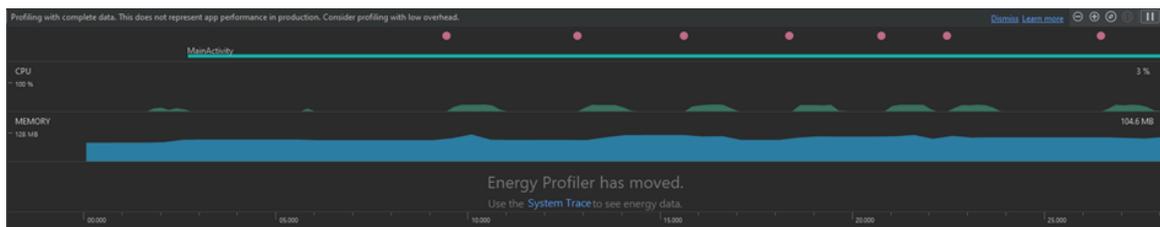
Dari Gambar 3 kita dapat melihat bahwa berdasarkan kinerja aplikasi yang menggunakan bahasa tersebut, dapat dilihat penggunaan CPU (atas/hijau) dan memori (bawah/biru). CPU hanya bekerja saat proses sedang berjalan. Proses yang dimaksud memanggil fungsi Iteration() dengan menekan tombol integer hingga hasilnya ditampilkan pada layar aplikasi. Mengenai penggunaan memori, memori tetap berfungsi meskipun tidak ada proses yang berjalan [1].



Gambar 2 Hasil tampilan dari kedua aplikasi

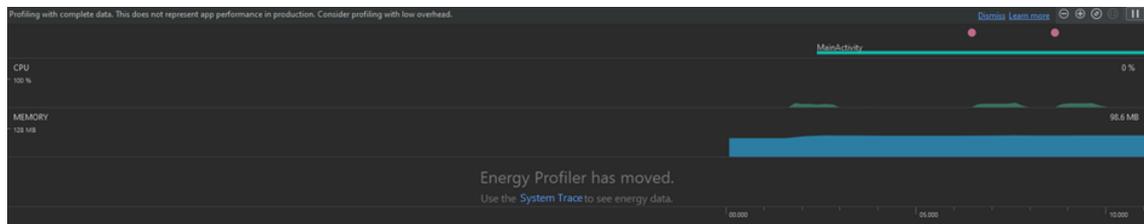


Gambar 3 CPU dan Memory String Java 10 detik



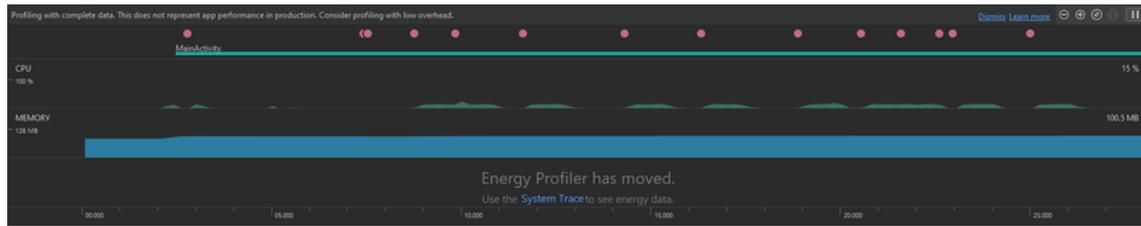
Gambar 4 CPU dan Memory String Java 25 detik

Dari Gambar 4, terlihat bahwa String Java selama 25 detik memiliki pola yang serupa dengan yang dalam 10 detik, hanya berlangsung dalam rentang waktu yang lebih panjang. Penelitian ini dimaksudkan untuk menganalisis dampak durasi eksekusi terhadap pola dan kinerja pengolahan string di Java. Dengan membandingkan hasil pada durasi yang berbeda, penelitian ini bertujuan untuk mengevaluasi efisiensi algoritma, stabilitas kinerja, serta perubahan pola yang mungkin terjadi seiring berjalannya waktu proses. Hasil temuan ini diharapkan dapat memberikan kontribusi dalam mengoptimalkan algoritma pengolahan string untuk skenario eksekusi dalam jangka waktu yang lebih panjang.



Gambar 5 CPU dan Memory Integer Java

Dari Gambar 5, terlihat bahwa CPU aktif selama proses berlangsung, sementara penggunaan memori tetap konstan seperti pada pengujian string. Penelitian ini bertujuan untuk menganalisis penggunaan sumber daya perangkat, terutama CPU dan memori, selama eksekusi proses. Tujuan utamanya adalah mengevaluasi efisiensi pemrosesan, stabilitas penggunaan memori, dan beban kerja CPU. Diharapkan hasil ini dapat memberikan wawasan tentang manajemen sumber daya yang lebih baik serta membantu dalam mengoptimalkan kinerja aplikasi.



Gambar 6 CPU dan Memory Integer Java

Dalam Gambar 6, terlihat bahwa hasil pengujian menunjukkan pola serupa dengan pengujian selama 10 detik. Saat pengujian ini berlangsung, CPU hanya aktif selama proses berjalan, sementara memori tetap stabil. Penelitian ini dilakukan untuk menganalisis pola penggunaan CPU dan memori selama proses eksekusi. Tujuan utamanya adalah untuk mengevaluasi efisiensi pengelolaan sumber daya sistem, terutama dalam hal penggunaan CPU dan stabilitas memori. Diharapkan bahwa hasil penelitian ini dapat memberikan pemahaman yang lebih mendalam tentang kinerja aplikasi dan membantu dalam meningkatkan efisiensi proses eksekusi.

V. PEMBAHASAN

Tes kinerja string dan integer di Java dan Kotlin dilakukan selama 10 detik dengan 5 iterasi. Hasil dari tes tersebut telah disajikan dalam Tabel 3 hingga 6. Berikut adalah interpretasi dari hasilnya.

TABEL 3
 TESTING STRING KOTLIN 10 DETIK

Test Ke	Waktu (Nanodetik)	Total (Karakter)	CPU (%)		Memory (MB)	
			Min	Max	Min	Max
1	10.229.452.027	52.394	5	22	121,6	192,6
2	10.236.110.309	53.250	3	22	126,4	220,3
3	10.255.252.184	53.117	3	20	126,4	192,2
4	10.239.196.767	52.773	5	19	124,2	200,8
5	10.271.396.715	53.199	5	20	106,2	180
Rata-rata	10.246.281.600	52.947	4,2	20,6	120,96	197,18

TABEL 4
 TESTING STRING JAVA 10 DETIK

Test Ke	Waktu (Nanodetik)	Total (Karakter)	CPU (%)		Memory (MB)	
			Min	Max	Min	Max
1	10.336.234.683	69.965	8	33	120,7	270,3
2	10.324.212.392	70.551	5	30	125,9	265,4
3	10.326.921.090	70.615	4	28	155,3	228,6
4	10.333.519.267	70.272	9	27	150,3	230,9
5	10.325.802.079	70.324	9	26	130,3	295,9
Rata-rata	10.329.337.902	70.345	7,0	28,8	336,5	258,22

Berdasarkan Tabel 3 dan Tabel 4, dapat disimpulkan bahwa bahasa pemrograman Kotlin lebih baik dalam hal waktu eksekusi, terutama bila dibandingkan dengan bahasa pemrograman Java. Rata-rata waktu eksekusi dari pemrograman Kotlin adalah 10.246 ms, sedangkan pemrograman Java dapat mencapai waktu rata-rata eksekusi di angka 10.329 ms. Dengan kata lain, rata-rata pemrograman Kotlin lebih cepat sekitar 0,8% dalam melakukan proses pengolahan string dibandingkan dengan Java.

Tetapi, bila dilihat dari total jumlah karakter yang diolah, maka bahasa pemrograman Java lebih baik, dengan jumlah rata-rata 70.345 karakter, sedangkan bahasa pemrograman Kotlin hanya mampu memproses 52.947 karakter. Ini berarti walaupun waktu eksekusi dari Kotlin lebih cepat, Java mempunyai waktu karakter yang lebih besar.

Dari efisiensi dalam penggunaan sumber daya, ternyata melakukan pengolahan string dengan menggunakan bahasa pemrograman Kotlin lebih efisien. Pada rata-rata persentase penggunaan CPU Kotlin antara 4,2% - 20,6% sedangkan Java 7,0% - 28,8%. Sebaliknya, penggunaan memori oleh Kotlin berada pada nilai rata-rata 197,18 MB atau lebih rendah dibandingkan dengan Java yang mencapai 258,22 MB. Data ini menunjukkan bahwa pemrograman Kotlin lebih baik dalam melakukan manajemen string.

TABEL 5
 TESTING INTEGER JAVA 10 DETIK

Test Ke	Waktu (Nanodetik)	Total (Karakter)	CPU (%)		Memory (MB)	
			Min	Max	Min	Max
1	10.001.816.975	73.496.941	5	15	110,7	110,2
2	10.001.910.309	73.504.949	5	16	110,9	112,4
3	10.001.910.309	73.491.112	4	15	108,8	111,9
4	10.001.947.288	73.524.173	6	14	109,5	115,6
5	10.002.106.871	73.572.208	4	15	105	106,8
Rata-rata	10.001.901.975	73.517.877	4,8	15	108,98	111,38

TABEL 6
 TESTING INTEGER KOTLIN 10 DETIK

Test Ke	Waktu (Nanodetik)	Total (Karakter)	CPU (%)		Memory	
			Min	Max	Min	Max
1	10.007.478.590	72.932.489	5	15	110,7	115
2	10.004.526.715	73.805.665	4	14	100,1	110,9
3	10.004.489.892	73.849.689	4	16	112,8	111,7
4	10.005.471.506	73.841.621	2	15	112,6	112,8
5	10.003.665.309	73.946.750	4	15	115,4	114,5
Rata-rata	10.005.126.402	73.675.243	3,8	15	110,32	112,98

Dalam kinerja pemrograman berbasis komputasi menunjukkan penggunaan bahasa Tritunggal Bahasa seperti diinfokan Tabel 5 dan Tabel 6. Untuk menguji kinerja Java dilakukan satu set pengujian, hasil uji didapatkan waktu eksekusi rata – rata untuk Java adalah 10.002 ms, lebih cepat dari pada Kotlin yang time out 10.005 ms. Perbedaan ini relatif kecil sekitar 0,03% diuntungkan untuk Java. Namun dari segi perolehan jumlah iterasi yang dapat dilakukan adalah besarnya 73,675,243 iterasi ke-63: di Java dihanya 73,517,877 iterasi ke-62. Roti yangan bumbu km, menginginkan, meskipun, sebagian besar waktu eksekusi lebih pendek Plaintiff, Keiatam lebih besar diproses yaitu faktor.

Dari segi penggunaan lingkup sistem juga nampaknya Kotlin lebih unggul. Penggunaan CPU Kotlin pahami 3,8% sampai pada 15% sedangkan Java memiliki komposisi CPU 4,8 % hinggna 15%. Mediak dtype dword 4 huruf +93 nad 4 data Dengan cara menjumlah pemakaian dipakai sasaran mengeras, bubuh ditenga penggunaan memori didapatkan. Bagi kedua pemakai meskipun tidak signifikan bedanya, hal ini menunjukkan bahwa kedua bahasa tersebut cenderung memiliki pola pemakaian sumber daya yang mirip dalam pengolahan data integer.

Berdasarkan layanan analitik yang disediakan dalam Laporan seperti Data Analyst, sangat penting untuk mengenali batasannya seperti fakta bahwa tes dilakukan hanya dalam 5 iterasi dalam 10 detik yang mungkin berbeda dengan durasi pengecatan Jira Server dan atau iterasi; lingkungan eksekusi yang tergantung pada spesifikasi perangkat keras, sistem operasi, dan versi Java dan Kotlin yang digunakan; pembatasan pada jenis data yang hanya boleh dianggap string dan integer sementara jenis data lainnya dapat berkontribusi.

Untuk memperkuat hasil dan memberikan wawasan yang lebih mendalam, penelitian di masa depan dapat mempertimbangkan beberapa rekomendasi, yaitu memperpanjang durasi dan iterasi pengujian agar kinerja yang dihasilkan lebih stabil dan akurat serta mampu mengungkap potensi anomali yang mungkin tidak terlihat dalam pengujian singkat. Selain itu, pengujian dengan berbagai jenis data, seperti float, double, array, dan objek kompleks, perlu dilakukan untuk mengevaluasi kinerja kedua bahasa secara lebih komprehensif. Lingkungan pengujian yang beragam, mencakup berbagai perangkat keras dan sistem operasi seperti Windows, macOS, dan Linux, juga penting agar kesimpulan yang dihasilkan lebih general dan aplikatif. Disarankan pula untuk melakukan pengujian dengan beban kerja nyata, seperti aplikasi berbasis API, pengolahan file besar, atau operasi database, sehingga hasil pengujian lebih relevan dengan kondisi dunia nyata. Selanjutnya, pengujian kinerja dengan beban multithreading atau pengolahan asynchronous akan memberikan gambaran yang lebih realistis, mengingat banyak aplikasi modern memanfaatkan pola pengolahan paralel. Dampak pengelolaan memori oleh garbage collector terhadap pengolahan string dan integer juga perlu diukur, sehingga pengaruhnya terhadap waktu eksekusi dapat dipahami dengan lebih baik. Terakhir, variasi pengukuran kinerja perlu ditingkatkan dengan menambahkan metrik seperti latensi, throughput, dan jumlah alokasi memori, sehingga analisis yang dihasilkan menjadi lebih lengkap dan informatif.

VI. KESIMPULAN

Penelitian tentang perbandingan performa bahasa pemrograman Kotlin dan Java menunjukkan hasil yang menarik. Kedua bahasa memiliki keunggulan dan kelemahan yang berbeda tergantung pada jenis data, konteks penggunaan, dan aspek kinerja yang diukur. Dalam hal waktu eksekusi pada pengolahan data numerik, Java terbukti lebih unggul. Itu berarti Java lebih cocok untuk proyek yang memerlukan pemrosesan bilangan bulat dalam jumlah besar, seperti aplikasi berbasis perhitungan ilmiah atau sistem keuangan yang membutuhkan kecepatan eksekusi tinggi. Namun, Kotlin memiliki keunggulan dalam pemrosesan string. Bahasa ini lebih efisien dalam penggunaan CPU, memori, dan pengelolaan sumber daya. Oleh karena itu, Kotlin menjadi pilihan yang sesuai untuk proyek yang fokus pada efisiensi pengelolaan data string. Contoh proyek-proyek yang cocok dengan Kotlin adalah aplikasi chat, sistem pencarian teks, dan pengelolaan konten dinamis. Dalam memilih bahasa pemrograman yang tepat, penting untuk mempertimbangkan kebutuhan proyek dan tujuan yang ingin dicapai. Evaluasi lebih lanjut terkait konteks penggunaan dan aspek kinerja yang diukur akan memberikan pemahaman yang lebih mendalam tentang perbandingan antara Kotlin dan Java.

Penelitian ini menyediakan panduan komprehensif bagi pengembang dalam memilih bahasa pemrograman yang sesuai untuk proyek mereka. Rekomendasi Java untuk pengembangan aplikasi berbasis data numerik dan proyek yang membutuhkan eksekusi cepat, seperti sistem analitik, perangkat lunak simulasi, dan aplikasi berbasis IoT. Di sisi lain, Kotlin lebih disarankan untuk aplikasi yang memerlukan efisiensi dalam penggunaan sumber daya dan pengelolaan string yang kompleks, seperti aplikasi media sosial, aplikasi berbasis API, dan pengelolaan antarmuka pengguna yang dinamis. Penelitian ini juga berkontribusi pada literatur yang membahas kinerja aplikasi Android dengan rekomendasi pengujian yang melibatkan algoritma yang lebih kompleks dan dataset uji yang lebih besar serta beragam. Tujuannya adalah memberikan pemahaman yang lebih komprehensif mengenai efisiensi dan keandalan kedua bahasa dalam konteks pengembangan aplikasi di dunia nyata.

Bagi pengembang aplikasi di Android Studio, Java sering dipilih oleh pemula karena stabilitas dan popularitasnya yang luas. Meskipun demikian, Kotlin merupakan alternatif menarik berkat sintaksis yang ringkas, fitur null safety, dan pengurangan kode boilerplate. Pengembang yang sudah terbiasa dengan Java dapat dengan mudah beradaptasi dengan Kotlin. Karenanya, disarankan untuk memulai dengan belajar Java terlebih dahulu sebelum beralih ke Kotlin guna meraih efisiensi pengelolaan kode dan kemudahan pengembangan aplikasi modern. Keputusan yang bijak dalam memilih bahasa pemrograman dapat meningkatkan produktivitas pengembang sekaligus menghasilkan aplikasi yang lebih efisien dalam kinerja dan penggunaan sumber daya.

DAFTAR PUSTAKA

- [1] G. C. Wicaksono, H. Leong, P. Studi, T. Informatika, and F. Ilmu, "JAVA DAN KOTLIN PADA ANDROID APPS," vol. 4, no. 2, pp. 85–96, 2021.
- [2] M. A. Alkandari, A. Kelkawi, and M. O. Elish, "An Empirical Investigation on the Effect of Code Smells on Resource Usage of Android Mobile Applications," *IEEE Access*, vol. 9, pp. 61853–61863, 2021, doi: 10.1109/ACCESS.2021.3075040.
- [3] L. Ardito, R. Coppola, G. Malnati, and M. Torchiano, "Effectiveness of Kotlin vs. Java in android app development tasks," *Inf. Softw. Technol.*, vol. 127, p. 106374, Nov. 2020, doi: 10.1016/j.infsof.2020.106374.
- [4] L. Damingo, K. N. Elliot, L. A. S. Damingo, and K. N. Elliot, "A Comparative Review of Mobile Application Development Frameworks: Kotlin Vs Java," 2022. [Online]. Available: <https://www.researchgate.net/publication/363721478>
- [5] H. Nakamura, A. Sonoyama, T. Kamiyama, M. Oguchi, and S. Yamaguchi, "Performance Study of Kotlin and Java Programs with Bytecode Analysis," *J. Inf. Process.*, vol. 32, no. 0, pp. 380–395, 2024, doi: 10.2197/ipsjip.32.380.
- [6] D. Sulowski and G. Koziel, "Analiza porównawcza języków Kotlin i Java używanych do tworzenia aplikacji na system Android," *J. Comput. Sci. Inst.*, vol. 13, pp. 354–358, 2019.
- [7] A. Febriandirza, "Perancangan Aplikasi Absensi Online Dengan Menggunakan Bahasa Pemrograman Kotlin," *Pseudocode*, vol. 7, no. 2, pp. 123–133, Sep. 2020, doi: 10.33369/pseudocode.7.2.123-133.
- [8] S. M. Khan, A. U. Nabi, and T. H. Bhanbhro, "Comparative Analysis between Flutter and React Native," *Int. J. Artif. Intell. Math. Sci.*, vol. 1, no. 1, pp. 15–28, Sep. 2022, doi: 10.58921/ijaims.v1i1.19.
- [9] S.-H. Lee, S.-H. Kim, J. Y. Hwang, S. Kim, and S.-H. Jin, "Is Your Android App Insecure? Patching Security Functions With Dynamic Policy Based on a Java Reflection Technique," *IEEE Access*, vol. 8, pp. 83248–83264, 2020, doi: 10.1109/ACCESS.2020.2987059.
- [10] K. Wasilewski and W. Zabierowski, "A Comparison of Java, Flutter and Kotlin/Native Technologies for Sensor Data-Driven Applications," *Sensors*, vol. 21, no. 10, p. 3324, May 2021, doi: 10.3390/s21103324.
- [11] D. Martinez, X. Ferre, G. Guerrero, and N. Juristo, "An Agile-Based Integrated Framework for Mobile Application Development Considering Ilities," *IEEE Access*, vol. 8, pp. 72461–72470, 2020, doi: 10.1109/ACCESS.2020.2987882.
- [12] S. N. S. Muslim, F. Nurdiansyah, and A. Y. Rahman, "PERBANDINGAN ALGORITMA NAIVE BAYES DAN KNN DALAM ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI CAPCUT," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3S1, pp. 61–69, Oct. 2024, doi: 10.23960/jitet.v12i3S1.5156.
- [13] R. Sikder, M. S. Khan, M. S. Hossain, and W. Z. Khan, "A survey on android security: development and deployment hindrance and best practices," *TELKOMNIKA (Telecommunication Comput.*
- [14] K. Java, N. Everlönn, and S. Gakis, "Independent project, 15 credits, for the degree of Bachelor of Computer Science Spring Semester 2020 Faculty of Natural Sciences."
- [15] F. Elyana, "Kajian Perbandingan Penggunaan Bahasa Java dan Kotlin dalam Android Studio," *e-Proceedings 11' Natl. Conf. Educ. Tech. Vocat. Educ. Train. 2021 JPPKK Politek. Sultan Azlan Shah*, vol. 0, 2021.