

Comparative Analysis Retrofit and Ktor Client Performance in Various Internet Speeds Internet on MSMEs Cashier Application

Muhamad Akbar Abdul Kholik^{*1}, Dinar Nugroho Pratomo²

^{1,2}Gadjah Mada University, Yogyakarta, Indonesia

Email: ¹muhamadakbar2020mail.ugm.ac.id

Received : Oct 11, 2024; Revised : Nov 4, 2024; Accepted : Nov 5, 2024; Published : Jun 10, 2025

Abstract

MSMEs (Micro, Small, and Medium Enterprises) in Indonesia face uneven network infrastructure, with more than 20% of smartphone users having download speeds below 10 Mbps. This condition hampers the efficiency of data processing between client and server, while MSMEs need innovations such as digitization of bookkeeping to increase competitiveness. The selection of HTTP networking libraries such as Retrofit and Ktor Client is very important, because both play a role in the process of sending and receiving data from the server. This research aims to analyze the performance of both libraries in the Lulu POS application to determine the most optimal library in supporting MSME operations in various network conditions. The test is conducted in two scenarios: the first scenario uses text data and the second scenario uses text and image data. Each scenario has several test cases that will be tested at six different internet speeds. The results show that Retrofit excels in response time for text data with a performance improvement of 18.85% and network usage of 21.33%. Ktor Client is superior in scenarios involving text and image data, with a response time advantage of 7.20% and network usage of 0.08%. On the other hand, Retrofit is more efficient in memory usage in both scenarios, with an advantage of 16.49% in text data and 4.70% in text and image data. In conclusion, Retrofit is more stable for applications focusing on text data such as Lulu POS, while Ktor Client is more suitable for applications that manage images. These results make MSMEs get cashier applications with optimal libraries for various network conditions, so that operations are smoother and data management efficiency increases.

Keywords: HTTP networking library, Ktor Client, Retrofit

This work is an open access article and licensed under a Creative Commons Attribution-Non Commercial 4.0 International License



1. PENDAHULUAN

UMKM (Usaha Mikro Kecil dan Menengah) adalah kegiatan usaha produktif yang dijalankan oleh individu atau badan usaha di berbagai sektor, mulai dari perdagangan hingga jasa, dengan tujuan menghasilkan keuntungan dan mendorong pertumbuhan ekonomi. Potensi UMKM dalam menggerakkan perekonomian harus mendapatkan perubahan serta inovasi untuk meningkatkan daya saing mereka [1]. Salah satu inovasi penting yang dapat mendukung perkembangan UMKM adalah penerapan pembukuan digital [2].

Pembukuan digital memiliki peran penting untuk keberlanjutan UMKM, karena kesalahan dalam pengelolaan keuangan dapat menyebabkan ketidakstabilan uang kas, yang berpotensi membuat UMKM mengalami kerugian [3]. Dalam hal ini aplikasi kasir berperan penting untuk membantu pelaku UMKM dalam meningkatkan produktivitas, mengoptimalkan pengalaman pelanggan, serta mendukung pengambilan keputusan yang lebih cepat dan tepat untuk pertumbuhan bisnis mereka [4].

Namun, salah satu tantangan yang dihadapi oleh UMKM dalam mengadopsi teknologi ini adalah kondisi infrastruktur jaringan yang belum merata di Indonesia [5]. Menurut [6], rata-rata kecepatan

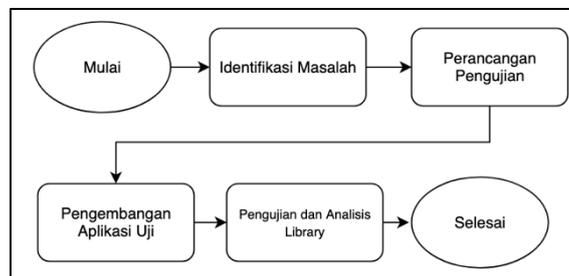
internet pengguna *smartphone* di Indonesia bervariasi, dengan 20,4% memiliki kecepatan 0-10 Mbps, 32% memiliki kecepatan 10-20 Mbps, 22,2% memiliki kecepatan 20-30 Mbps, dan 12,8% memiliki kecepatan di atas 40 Mbps. Kondisi ini menimbulkan tantangan dalam memastikan aplikasi kasir tetap dapat berfungsi dengan baik, terutama pada pengguna dengan koneksi internet yang lambat. Ketika aplikasi berjalan dengan lambat, pengguna dapat merasa tidak nyaman dan dapat menghilangnya minat dan keinginan pengguna untuk menggunakan aplikasi yang berpotensi menimbulkan kerugian bagi perusahaan [7].

Oleh karena itu, diperlukan solusi yang dapat meningkatkan efisiensi dan kecepatan pemrosesan data untuk memastikan UMKM tetap dapat memanfaatkan teknologi dengan efektif, meskipun dalam kondisi jaringan yang terbatas. Pemilihan HTTP *networking library* yang tepat sangat penting, terutama bagi aplikasi yang menghadapi keterbatasan kecepatan internet. *Library* yang optimal harus mampu menangani permintaan jaringan secara efisien dan cepat, bahkan pada koneksi internet yang lambat. Penggunaan *library* yang sesuai dapat meningkatkan performa aplikasi, sehingga UMKM tetap dapat beroperasi lancar dalam berbagai kondisi jaringan di Indonesia.

Penelitian ini bertujuan untuk membandingkan performa dua HTTP *networking library*, yaitu Retrofit dan Ktor Client, dalam pengembangan aplikasi Lulu POS sebagai studi kasus. Penelitian ini menjadi sangat penting karena aplikasi Lulu POS dirancang untuk digunakan oleh UMKM yang mungkin menghadapi kondisi jaringan yang buruk. Hasil analisis diharapkan membantu pengembang memilih *library* yang paling optimal untuk menghadapi kendala jaringan sehingga aplikasi dapat berjalan lancar dan efektif mendukung UMKM dalam keterbatasan konektivitas.

2. METODE PENELITIAN

Penelitian ini menggunakan beberapa tahapan seperti pada Gambar 1, tahapan penelitian terdiri dari 4 tahapan utama, yaitu identifikasi masalah, perancangan pengujian, pengembangan aplikasi uji, serta pengujian dan analisis *library*.



Gambar 1. Tahapan Penelitian

2.1. Identifikasi Masalah

Pada mengidentifikasi permasalahan, peneliti melakukan kajian pustaka dengan melibatkan analisis terhadap studi-studi sebelumnya yang relevan dengan topik yang diteliti. Kajian pustaka ini bertujuan untuk memahami konteks dan latar belakang masalah yang dihadapi. Peneliti mengumpulkan informasi dari berbagai sumber, termasuk jurnal ilmiah, artikel, dan laporan penelitian, guna mendapatkan wawasan yang komprehensif. Dengan pendekatan ini, peneliti berharap dapat merumuskan hipotesis yang tepat dan mengidentifikasi faktor-faktor kunci yang berkontribusi terhadap permasalahan yang sedang diteliti. Hasil dari kajian pustaka ini akan menjadi dasar yang kuat untuk melanjutkan penelitian ini.

2.2. Perancangan Pengujian

Pada perancangan pengujian, penulis akan menyusun rancangan mengenai bagaimana pengujian akan dilakukan dan metrik apa saja yang akan dipakai. Kecepatan internet yang diuji meliputi 1 Mbps, 10 Mbps, 20 Mbps, 30 Mbps, 40 Mbps, dan 50 Mbps. Pemilihan ini didasarkan pada data rata-rata kecepatan internet pengguna di Indonesia, di mana 20,4% pengguna berada di bawah 10 Mbps, 32% pada 10-20 Mbps, 22,2% pada 20-30 Mbps, dan 12,8% di atas 40 Mbps [6]. Penggunaan berbagai kecepatan internet ini diharapkan dapat menyimulasikan pengalaman nyata pengguna di Indonesia,

sehingga dengan pengujian ini dapat mengetahui performa tiap *library* dalam menghadapi beragam situasi jaringan yang umum dihadapi di Indonesia.

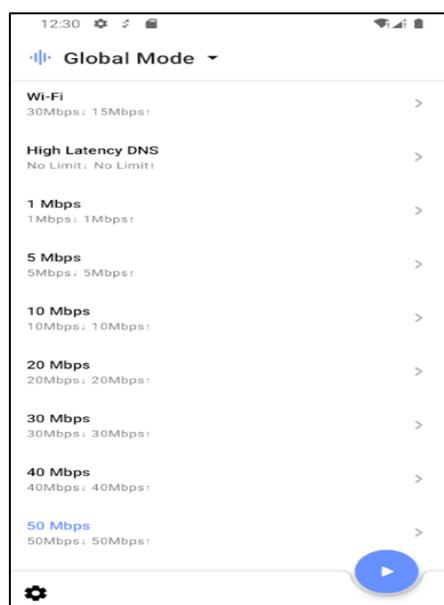
Skenario pengujian akan mencakup dua jenis kasus uji: yang pertama kasus uji menggunakan data teks serta yang kedua kasus uji menggunakan data teks dan gambar. Kasus uji ini disusun berdasarkan fitur-fitur yang tersedia di aplikasi Lulu POS, yang memproses kedua jenis data tersebut. Setiap kasus uji akan menjalani 20 kali percobaan untuk mendapatkan variasi nilai [8]. Selanjutnya, metrik evaluasi performa HTTP *networking library* meliputi *response time*, *network usage*, dan *memory usage*. Penentuan metrik ini berdasarkan standar ISO/IEC 25010:2023, yang mana dalam aspek *performance efficiency* menekankan bahwa aplikasi harus memenuhi kriteria efisiensi terkait *time behavior* dan *resource utilization* [9]. *Response time* mewakili *time behavior*, mengukur kecepatan aplikasi dalam merespons permintaan pengguna. *Network usage* merepresentasikan *resource utilization* terkait efisiensi penggunaan *bandwidth*. Sementara itu, *memory usage* juga termasuk dalam *resource utilization*, berfokus pada efisiensi penggunaan memori untuk menjaga stabilitas aplikasi. Ketiga metrik ini memberikan gambaran tentang efisiensi dan keandalan aplikasi sesuai standar *performance efficiency*.

2.2.1. HTTP Networking Library

HTTP *networking library* adalah sebuah kumpulan kode dan fungsi yang memungkinkan aplikasi berkomunikasi dengan server menggunakan protokol HTTP [10]. *Library* ini penting untuk mengelola pertukaran data antara aplikasi dan server, terutama pada aplikasi yang bergantung pada koneksi internet untuk berbagai operasi. Penelitian ini membandingkan dua *library* utama, yaitu Retrofit dan Ktor Client. Retrofit dipilih karena pada beberapa penelitian sebelumnya terbukti unggul dalam berbagai aspek performa dan memiliki tingkat popularitas mencapai 16,46% di antara aplikasi lainnya [11]. Di sisi lain, Ktor Client dipilih karena belum ada penelitian yang secara khusus menganalisis performanya. Ktor Client merupakan produk dari JetBrains [12], perusahaan yang juga mengembangkan bahasa pemrograman Kotlin [13], menjadikannya menarik untuk diteliti lebih lanjut karena reputasi perusahaan tersebut dalam inovasi teknologi dan pengembangan perangkat lunak.

2.2.2. Bandwidth Limiter

Pada pengujian *library* akan dilakukan pada beberapa kecepatan internet yang sudah ditentukan. Oleh karena itu, diperlukan sebuah *bandwidth limiter*. Pada pengujian ini penulis akan menggunakan sebuah aplikasi bernama Throttly. Aplikasi ini dapat melakukan limitasi pada *bandwidth* yang tersedia melalui koneksi VPN. Berikut tampilan aplikasi Throttly ditampilkan pada Gambar 2.



Gambar 2. Aplikasi Throttly

2.2.3. Kasus Uji

Kasus uji yang digunakan pada penelitian ini terdapat dua jenis, yaitu kasus uji dengan data teks serta kasus uji dengan data teks dan gambar. Pembagian kasus uji ini berfungsi untuk mendapatkan wawasan bagaimana performa *library* dalam menangani berbagai jenis data. Detail kasus uji dengan data teks ditampilkan pada Tabel 1. Sedangkan untuk kasus uji dengan data teks dan gambar ditampilkan pada

Tabel 2. Setiap tabel menyajikan informasi mengenai *request size*, yaitu ukuran data yang dikirim dari aplikasi ke server; *response size*, yang menggambarkan ukuran data yang diterima dari server; serta *total*, yang merupakan jumlah dari *request size* dan *response size*.

Tabel 1. Kasus Uji dengan Data Teks

No.	Kasus Uji	<i>Request Size</i> (byte)	<i>Response Size</i> (byte)	Total (byte)
1.	<i>Login</i>	445	950	1395
2.	<i>Register</i>	942	259	1201
3.	<i>Get 5 Product</i>	203	16445	16648
4.	<i>Get 50 Product</i>	204	161146	161350
5.	<i>Delete Product</i>	200	357	557
6.	<i>Get 5 Transaction</i>	254	51456	51710
7.	<i>Get 50 Transaction Create</i>	350	511508	511858
8.	<i>Transaction 5 Item Product Create</i>	1157	16814	17971
9.	<i>Transaction 50 Item Product</i>	7055	160256	167311

Tabel 2. Kasus Uji dengan Data Teks dan Gambar

No.	Kasus Uji	<i>Request Size</i> (byte)	<i>Response Size</i> (byte)	Total (byte)
1.	<i>Create Product 500 KB Image</i>	513177	656	513833
2.	<i>Create Product 5 MB Image Update</i>	5306368	656	5307024
3.	<i>Product 500 KB Image Update</i>	513198	3594	516792
4.	<i>Product 5 MB Image</i>	5306368	3594	5309962

2.2.4. Pengujian *Response Time*

Pengujian *Response Time* adalah pengujian dengan mengukur kecepatan sebuah *library* dalam memproses HTTP *request* hingga menerima *response* dari *server* [8]. Pengujian ini dilakukan dengan mencatat waktu awal saat pengujian dimulai hingga proses selesai. Kode pengujian *response time* ditunjukkan pada Gambar 3.

```
1 val startTime = System.currentTimeMillis()
2 val endTime = System.currentTimeMillis()
3 val elapsedTime = endTime - startTime;
```

Gambar 3. Kode Pengujian *Response Time*

2.2.5. Pengujian *Network Usage*

Pada pengujian *network usage* akan menghitung total *network usage* (Receive + Transmit) saat *library* memulai HTTP *request* hingga selesai. Pengujian ini akan menggunakan *class TrafficStats* yang berfungsi untuk mengambil total *network usage* berdasarkan UID aplikasi [14]. Kode pengujian *network usage* ditunjukkan pada Gambar 4.

```
1 val startRxBytes = TrafficStats.getUidRxBytes(android.os.Process.myUid())
2 val startTxBytes = TrafficStats.getUidTxBytes(android.os.Process.myUid())
3
4 val endRxBytes = TrafficStats.getUidRxBytes(android.os.Process.myUid())
5 val endTxBytes = TrafficStats.getUidTxBytes(android.os.Process.myUid())
6
7 val totalTxBytes = (endTxBytes - startTxBytes)
8 val totalRxBytes = (endRxBytes - startRxBytes)
9
10 val totalUsage = totalTxBytes + totalRxBytes
```

Gambar 4. Kode Pengujian *Network Usage*

2.2.6. Pengujian *Memory Usage*

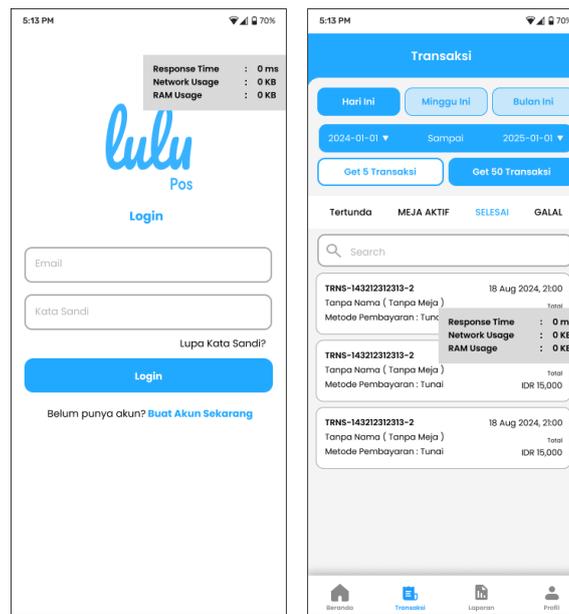
Pengujian *memory usage* adalah pengujian dengan mengambil penggunaan RAM pada saat *library* berhasil melakukan HTTP *request*. Pengujian ini akan menggunakan *function getTotalPrivateDirty()* yang ada pada *class Debug.MemoryInfo()*. *Function* ini berfungsi untuk mengambil jumlah memory yang digunakan oleh suatu proses tanpa dipengaruhi oleh proses lainnya [15]. Kode pengujian *memory usage* ditunjukkan pada Gambar 5.

```
1 val memInfo = Debug.MemoryInfo()
2 Debug.getMemoryInfo(memInfo)
3 val totalPrivateDirty = memInfo.totalPrivateDirty.toLong()
```

Gambar 5. Kode Pengujian *Memory Usage*

2.3. Perancangan Aplikasi Uji

Setelah merancang pengujian, langkah selanjutnya adalah merancang aplikasi uji yang akan digunakan dalam proses pengujian. Pada tahap ini akan dilakukan rancangan antarmuka dan arsitektur yang dipakai. Rancangan antarmuka akan melakukan modifikasi pada antarmuka yang sudah tersedia, modifikasi dilakukan untuk mempermudah dalam proses pengujian. Rancangan antarmuka ditampilkan pada Gambar 6. Selanjutnya arsitektur yang dipakai adalah MVVM (*Model, View, ViewModel*) di mana pada arsitektur ini memisahkan antara proses bisnis dan UI [16]. Dengan arsitektur ini, proses pengembangan aplikasi uji menjadi lebih mudah, karena pada pengembangan aplikasi uji, dua *library* berbeda akan digunakan, sehingga perpindahan *library* hanya memerlukan perubahan pada proses bisnisnya, tanpa perlu mengubah antarmuka pengguna (UI).



Gambar 6. Rancangan Antarmuka

2.4. Pengembangan Aplikasi Uji

Setelah rancangan pengujian dan pengembangan aplikasi uji selesai dibuat, tahap selanjutnya adalah mengembangkan aplikasi uji sesuai rancangan. Aplikasi ini akan menggunakan bahasa pemrograman Kotlin dan arsitektur MVVM. Karena aplikasi uji akan mengimplementasikan dua HTTP *networking library* yang berbeda, dua *branch* akan dibuat di Git untuk memisahkan implementasi masing-masing *library*. Rancangan antarmuka diimplementasikan sesuai dengan rancangan yang telah dibuat.

2.5. Pengujian dan Analisis Library

Pada tahap ini, pengujian akan dilakukan sesuai rancangan pengujian, di mana pengujian akan dilakukan pada enam kecepatan internet, yaitu 1 Mbps, 10 Mbps, 20 Mbps, 30 Mbps, 40 Mbps, 50 Mbps. Selain itu pengujian akan dilakukan pada dua jenis kasus uji yaitu kasus uji dengan data teks serta kasus uji dengan data teks dan gambar. Tiap kasus uji akan dilakukan pengujian sebanyak 20 kali percobaan untuk mendapatkan variasi nilai.

Setelah pengujian selesai dan hasilnya diperoleh, akan dilakukan evaluasi performa *library* berdasarkan berbagai metrik, yaitu *response time*, *network usage*, dan *memory usage*. Hasil dari setiap *library* kemudian akan dibandingkan untuk menentukan *library* mana yang memiliki performa lebih baik di berbagai kondisi jaringan.

3. HASIL DAN PEMBAHASAN

Penulis telah melaksanakan serangkaian pengujian untuk mengevaluasi kinerja dua HTTP *networking library*, yaitu Retrofit dan Ktor Client, dalam konteks aplikasi Lulu POS. Pengujian ini mencakup pengukuran *response time*, *network usage*, dan *memory usage* dalam berbagai kondisi jaringan.

3.1. Implementasi Aplikasi Uji

Implementasi aplikasi uji telah berhasil dilakukan sesuai rancangan aplikasi uji. Aplikasi sudah memiliki dua versi yang mengguna *library* yang berbeda yaitu Retrofit dan Ktor Client, dan sudah menerapkan arsitektur MVVM (*Model, View, ViewModel*) yang memisahkan logika bisnis dari antarmuka pengguna. Tampilan hasil implementasi aplikasi uji ditunjukkan pada Gambar 7.



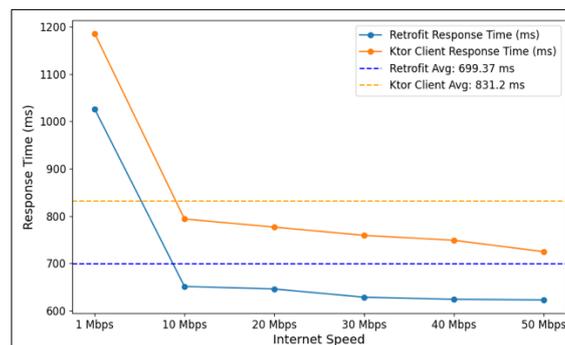
Gambar 7. Hasil Implementasi Aplikasi Uji

3.2. Hasil Pengujian *Response Time*

Pada bagian ini akan diuraikan hasil pengujian *response time* yang telah dilakukan untuk mengetahui performa kedua *library* dalam berbagai kondisi jaringan dan jenis data.

3.2.1. Hasil Pengujian *Response Time* pada Kasus Uji Data Teks

Pada bagian ini menyajikan hasil pengujian *response time* untuk data teks antara Retrofit dan Ktor Client pada berbagai kecepatan internet.

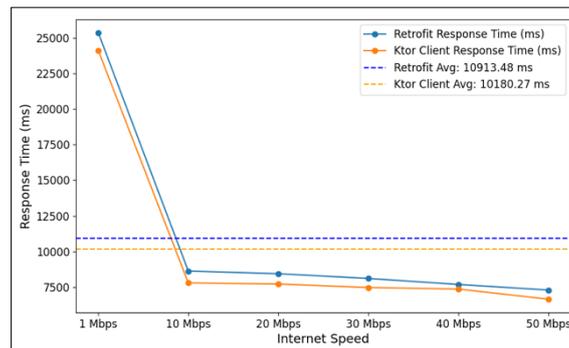


Gambar 8. Hasil Pengujian *Response Time* pada Data Teks

Hasil pengujian pada Gambar 8, menunjukkan bahwa Retrofit memiliki keunggulan dalam *response time* di seluruh kecepatan internet ketika menggunakan data teks. Rata-rata *response time* yang tercatat untuk Retrofit adalah 699,37 ms, sedangkan Ktor Client menunjukkan rata-rata *response time* yang lebih tinggi, yaitu 831,20 ms. Selisih rata-rata *response time* antara kedua *library* ini mencapai 131,84 ms, yang menunjukkan bahwa Retrofit lebih unggul sebesar 18,85%. Hal ini menunjukkan bahwa Retrofit memberikan kinerja *response time* yang lebih cepat dibandingkan dengan Ktor Client dalam pengujian yang menggunakan data teks.

3.2.2. Hasil Pengujian *Response Time* pada Teks dan Gambar

Pada bagian ini menyajikan hasil pengujian *response time* untuk data teks dan gambar antara Retrofit dan Ktor Client pada berbagai kecepatan internet.



Gambar 9. Hasil Pengujian *Response Time* dengan Data Teks dan Gambar

Hasil pengujian pada Gambar 9, menunjukkan bahwa Ktor Client unggul dalam *response time* di semua kecepatan internet saat menggunakan data teks dan gambar. Rata-rata *response time* untuk Retrofit tercatat sebesar 10913,48 ms, sedangkan Ktor Client lebih baik dengan rata-rata 10180,27 ms. Selisih rata-rata *response time* antara kedua *library* ini mencapai 733,21 ms, menunjukkan keunggulan Ktor Client sebesar 7,20%. Ini mengindikasikan bahwa Ktor Client memberikan kinerja *response time* yang lebih cepat dibandingkan Retrofit pada pengujian data teks dan gambar.

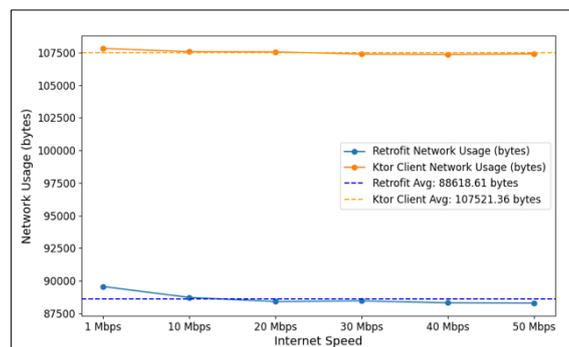
Selanjutnya dilakukan analisis terhadap hasil pengujian. Hasil analisis pada pengujian *response time* menunjukkan bahwa Retrofit unggul pada kasus uji dengan data teks sebesar 18,85%, sedangkan Ktor Client memiliki keunggulan pada kasus uji dengan data teks dan gambar sebesar 7,20%. Perbedaan ini menunjukkan bahwa masing-masing *library* memiliki keunggulan tersendiri dalam pemrosesan jenis data yang berbeda. Secara keseluruhan, semua *library* menunjukkan penurunan *response time* pada kecepatan internet yang lebih tinggi. *Response time* yang cepat dapat berpengaruh positif terhadap pengalaman pengguna aplikasi kasir di lingkungan nyata, meningkatkan efisiensi transaksi dan kepuasan pelanggan.

3.3. Hasil Pengujian *Network Usage*

Selanjutnya, akan menjelaskan hasil pengujian *network usage* yang telah dilakukan untuk mengevaluasi kinerja kedua *library* dalam berbagai kondisi jaringan dan jenis data.

3.3.1. Hasil Pengujian *Network Usage* pada Data Teks

Pada bagian ini menyajikan hasil pengujian *network usage* untuk data teks antara Retrofit dan Ktor Client pada berbagai kecepatan internet.



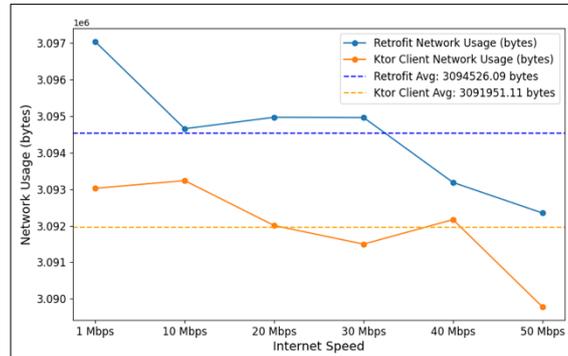
Gambar 10. Hasil Pengujian *Network Usage* dengan Data Teks

Hasil pengujian pada Gambar 10, menunjukkan bahwa Retrofit memiliki keunggulan dalam *network usage* di seluruh kecepatan internet ketika diuji dengan data teks. Rata-rata *network usage* yang tercatat untuk Retrofit adalah 88618,61 byte, sedangkan Ktor Client menunjukkan rata-rata *network usage* yang lebih tinggi, yaitu 107521,36 byte. Selisih rata-rata *network usage* antara kedua *library* ini mencapai 18902,75 byte, yang menunjukkan bahwa Retrofit lebih unggul sebesar 21,33%. Ini

menunjukkan bahwa Retrofit memberikan kinerja *network usage* yang lebih efisien dibandingkan Ktor Client dalam pengujian dengan data teks.

3.3.2. Hasil Pengujian *Network Usage* pada Data Teks dan Gambar

Pada bagian ini menyajikan hasil pengujian *network usage* untuk data teks dan gambar antara Retrofit dan Ktor Client pada berbagai kecepatan internet.



Gambar 11. Hasil Pengujian *Network Usage* dengan Data Teks dan Gambar

Hasil pengujian pada Gambar 11, menunjukkan bahwa Ktor Client memiliki keunggulan dalam *network usage* di semua kecepatan internet ketika diuji dengan data teks dan gambar, meskipun perbedaannya tidak signifikan. Rata-rata *network usage* yang tercatat untuk Retrofit adalah 3094526,09 byte, sedangkan Ktor Client mencatat rata-rata *network usage* yang lebih rendah, yaitu 3091951,11 byte. Selisih rata-rata *network usage* antara kedua *library* ini mencapai 2574,98 byte, yang menunjukkan bahwa Ktor Client lebih unggul sebesar 0,08%. Hal ini menunjukkan bahwa Ktor Client memberikan kinerja *network usage* yang lebih efisien dibandingkan dengan Retrofit dalam pengujian yang melibatkan data teks, meskipun selisihnya sangat kecil.

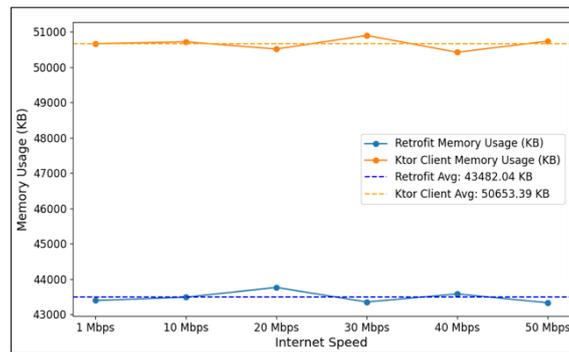
Selanjutnya dilakukan analisis terhadap hasil pengujian. Hasil analisis pada pengujian *network usage* menunjukkan bahwa Retrofit unggul pada kasus uji dengan data teks sebesar 21,33%, sementara Ktor Client memiliki keunggulan pada kasus uji dengan data teks dan gambar sebesar 0,08%. Perbedaan ini menunjukkan bahwa masing-masing *library* memiliki keunggulan tersendiri dalam pemrosesan jenis data yang berbeda. Secara keseluruhan, kedua *library* menunjukkan penurunan *network usage* pada kecepatan internet yang lebih tinggi, meskipun perubahan tersebut tidak terlalu signifikan. Dengan efisiensi *network usage*, dapat berpotensi mengurangi beban jaringan dan mengoptimalkan pengalaman pengguna, terutama saat aplikasi digunakan di wilayah dengan *bandwidth* terbatas atau biaya data yang tinggi.

3.4. Hasil Pengujian *Memory Usage*

Hasil pengujian terakhir adalah *memory usage*. Hasil yang telah didapat akan dilakukan analisis kinerja kedua *library* dalam berbagai kondisi jaringan dan jenis data.

3.4.1. Hasil Pengujian *Memory Usage* pada Data Teks

Pada bagian ini menyajikan hasil pengujian *memory usage* untuk data teks antara Retrofit dan Ktor Client pada berbagai kecepatan internet.

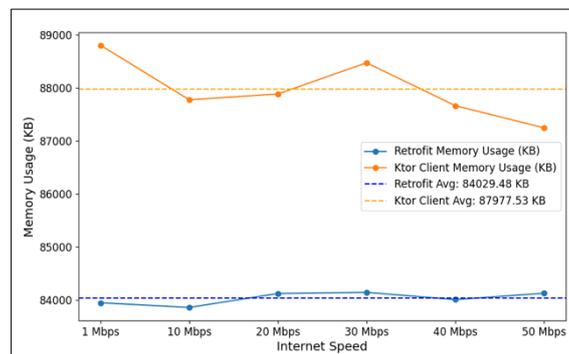


Gambar 12. Hasil Pengujian *Memory Usage* dengan Data Teks

Hasil pengujian pada Gambar 12, menunjukkan bahwa Retrofit memiliki keunggulan dalam *memory usage* di semua kecepatan internet ketika diuji dengan data teks. Rata-rata *memory usage* yang tercatat untuk Retrofit adalah 43482,04 KB, sedangkan Ktor Client menunjukkan rata-rata *memory usage* yang lebih tinggi, yaitu 50653,39 KB. Selisih rata-rata *memory usage* antara kedua *library* ini mencapai 7171,35 KB, yang menunjukkan bahwa Retrofit lebih unggul sebesar 16,49%. Ini menunjukkan bahwa Retrofit memberikan kinerja *memory usage* yang lebih baik dibandingkan dengan Ktor Client dalam pengujian yang melibatkan data teks.

3.4.2. Hasil Pengujian *Memory Usage* pada Data Teks dan Gambar

Pada bagian ini menyajikan hasil pengujian *memory usage* untuk data teks dan gambar antara Retrofit dan Ktor Client pada berbagai kecepatan internet.



Gambar 13. Hasil Pengujian *Memory Usage* dengan Data Teks dan Gambar

Hasil pengujian pada Gambar 13, menunjukkan bahwa Retrofit juga unggul dalam *memory usage* di semua kecepatan internet ketika diuji dengan data teks dan gambar, sebagaimana ditunjukkan pada Gambar 4.32. Rata-rata *memory usage* yang tercatat untuk Retrofit adalah 84.029,48 KB, sedangkan Ktor Client menunjukkan rata-rata *memory usage* yang lebih tinggi, yaitu 87.977,53 KB. Selisih rata-rata *memory usage* antara kedua *library* ini mencapai 3.948,05 KB, yang menunjukkan bahwa Retrofit lebih unggul sebesar 4,70%. Hal ini menunjukkan bahwa Retrofit memberikan kinerja *memory usage* yang lebih baik dibandingkan dengan Ktor Client dalam pengujian yang melibatkan data teks dan gambar.

Selanjutnya dilakukan analisis terhadap hasil pengujian. Hasil analisis dari pengujian *memory usage* menunjukkan bahwa Retrofit memiliki kinerja yang lebih baik dibandingkan dengan Ktor Client di semua kasus yang diuji. Dalam pengujian menggunakan data teks, Retrofit unggul sebesar 16,49%, sementara untuk pengujian dengan data teks dan gambar, keunggulannya adalah 4,70%. Analisis ini mengindikasikan bahwa Retrofit lebih efisien dalam mengelola *memory usage* dibandingkan Ktor Client di berbagai kondisi pengujian yang diterapkan. Perbedaan kinerja ini konsisten di seluruh rentang kecepatan internet yang diuji, yang menegaskan keunggulan Retrofit dalam efisiensi *memory usage* untuk aplikasi yang diuji. Namun, peningkatan kecepatan internet tidak memberikan dampak signifikan

terhadap fluktuasi *memory usage* pada kedua *library* ini. Hasil ini menunjukkan Retrofit dapat memungkinkan aplikasi untuk berjalan lebih ringan, mengurangi risiko lag atau *crash*, terutama pada perangkat dengan kapasitas memori terbatas. Hal ini dapat meningkatkan pengalaman pengguna secara keseluruhan, karena aplikasi mampu menangani proses yang lebih besar atau kompleks tanpa mengorbankan performa

4. DISKUSI

Berdasarkan hasil uji performa, diperoleh analisis terkait kinerja Retrofit dan Ktor Client pada tiga metrik utama: *response time*, *network usage*, dan *memory usage*. Retrofit menunjukkan keunggulan dalam hal *response time* dan *network usage* pada kasus uji data teks. *Response time* Retrofit lebih cepat, dan *network usage*-nya lebih efisien, menjadikannya pilihan yang optimal untuk aplikasi yang memerlukan kecepatan respons tinggi serta efisiensi dalam penggunaan jaringan. Di sisi lain, Ktor Client menunjukkan performa yang lebih baik pada kasus uji yang melibatkan data teks dan gambar, dengan *response time* yang cepat dan *network usage* yang sedikit lebih efisien dalam skenario tersebut.

Dalam metrik *memory usage*, Retrofit menunjukkan konsistensi yang lebih baik, baik saat memproses data teks maupun data teks dan gambar. Pengujian juga menunjukkan bahwa kecepatan internet memengaruhi *response time* dan *network usage* dari kedua *library*, terutama pada jaringan lambat yang memperpanjang waktu respons. *Network usage* cenderung menurun dengan meningkatnya kecepatan internet, meskipun tidak secara signifikan, namun pada metrik *memory usage* tetap stabil, walaupun pada kecepatan internet yang lambat.

Berdasarkan hasil pengujian dan analisis yang telah diperoleh, penulis akan membandingkan temuan ini dengan hasil penelitian sebelumnya yang memiliki fokus serupa. Pendekatan ini diharapkan dapat memberikan wawasan tambahan mengenai kinerja HTTP *networking library* yang diuji serta menegaskan relevansi dan kontribusi penelitian ini dalam konteks studi yang lebih luas. Penulis menemukan penelitian yang berjudul "Implementasi HTTP *Library* Retrofit, Volley, dan *URLConnection* dalam Akses Data" oleh Binaya Amary [17], pada penelitian tersebut memiliki fokus yang sama yaitu membandingkan performa HTTP *networking library*, namun *library* yang diuji adalah Retrofit, Volley, dan *URLConnection*. Pada penelitian ini memiliki kesamaan dalam pembagian kasus uji, di mana pada kasus uji dibagi menjadi dua, yaitu kasus uji dengan data teks dan data gambar. Meskipun terdapat perbedaan dalam pengujian data gambar, hasil penelitian ini menunjukkan bahwa Retrofit unggul dalam pengambilan data teks, sementara Volley lebih cepat dalam mengambil data gambar. Temuan ini mengindikasikan bahwa Retrofit juga lebih efektif dalam mengelola data teks, selaras dengan hasil penelitian penulis.

Pada penelitian yang berjudul "Studi Performa Android *Networking Library* antara Fast Android Network Library, Retrofit dan OkHttp" oleh Vernanda et al. [10], melakukan penelitian dengan fokus serupa. *Library* yang diuji adalah Fast Android Network Library, Retrofit dan OkHttp. Hasil dari penelitian ini adalah ini menunjukkan bahwa OkHttp memiliki keunggulan dalam *response time*, meskipun perbedaannya tidak begitu signifikan dibandingkan dengan Retrofit. Dalam hal *network usage*, ketiga *library* menunjukkan hasil yang hampir setara, tetapi OkHttp tetap tampil lebih baik. Di sisi lain, dalam aspek *memory usage*, Retrofit menunjukkan performa yang lebih baik. Pada penelitian ini menunjukkan Retrofit memiliki performa hampir sama dengan OkHttp namun OkHttp tetap diunggulkan, namun Retrofit dapat menjadi alternatif, hasil ini menunjukkan Retrofit menjadi *library* yang cukup unggul dibandingkan *library* lain seperti pada penelitian ini.

Pada penelitian-penelitian sebelumnya sudah terbukti bahwa Retrofit memiliki keunggulan pada aspek performa dan menangani data teks, namun untuk Ktor Client belum ada penelitian terkait performanya. Hasil pengujian ini menunjukkan performa Ktor Client itu lebih bagus ketika menangani data teks dan gambar. Dengan demikian, Retrofit lebih ideal untuk aplikasi yang banyak mengelola data teks, seperti aplikasi kasir yang mengutamakan pengolahan dan sinkronisasi data transaksi. Sebaliknya, Ktor Client lebih cocok untuk aplikasi yang memerlukan banyak pengunggahan data teks dan gambar, seperti aplikasi sosial media dan layanan penyimpanan *cloud*. Melihat kelebihan masing-masing *library*, kedua *library* ini mungkin dapat digunakan secara bersamaan dalam satu aplikasi, dengan memanfaatkan keunggulan spesifik masing-masing sesuai tipe data yang diproses. Pendekatan ini bisa meningkatkan efisiensi dan fleksibilitas bagi pengguna. Namun, perlu mempertimbangkan potensi peningkatan ukuran

aplikasi, yang mungkin berdampak pada keterbatasan penyimpanan atau kinerja pada perangkat pengguna.

4.1. KESIMPULAN

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa Retrofit menunjukkan keunggulan dalam hal *response time* dan *network usage* saat menangani data teks. Secara spesifik, *response time* Retrofit lebih cepat 18,85%, dan *network usage*-nya lebih efisien 21,33%, menjadikannya pilihan yang optimal untuk pemrosesan data teks dengan kecepatan respons yang tinggi dan penggunaan jaringan yang lebih efisien. Sebaliknya, Ktor Client memiliki keunggulan dalam kasus yang melibatkan data teks dan gambar, dengan *response time* yang lebih cepat sebesar 7,20% dan *network usage* yang lebih efisien sebesar 0,08%. Pada sisi lain, pengujian *memory usage* menunjukkan bahwa Retrofit secara konsisten lebih unggul, baik pada skenario data teks (16,49%) maupun data teks dan gambar (4,70%). Kecepatan internet juga terbukti mempengaruhi *response time* dan *network usage* dari kedua *library* ini, terutama pada jaringan lambat yang menyebabkan waktu respons lebih lama. *Network usage* cenderung menurun seiring dengan meningkatnya kecepatan internet, meskipun tidak secara signifikan, sedangkan *memory usage* tetap stabil dan tidak terpengaruh oleh kecepatan internet, melainkan dipengaruhi oleh bagaimana aplikasi mengelola dan memproses data. Berdasarkan analisis ini, Retrofit direkomendasikan sebagai pilihan terbaik untuk aplikasi Lulu POS, yang sebagian besar fiturnya (28 dari 30) berfokus pada pemrosesan data teks dengan sedikit kebutuhan untuk pengunggahan gambar. Performanya yang konsisten dalam *response time*, *network usage*, dan *memory usage* menjadikannya pilihan yang lebih sesuai dibandingkan Ktor Client untuk kebutuhan aplikasi ini.

Hasil penelitian ini menunjukkan bahwa pemilihan HTTP *networking library* memiliki dampak terhadap kinerja aplikasi kasir di Indonesia, terutama dalam konteks kecepatan internet yang bervariasi di berbagai daerah. Dengan merekomendasikan Retrofit sebagai pilihan utama untuk aplikasi yang banyak memproses data teks, pengembang dapat memastikan pengalaman pengguna yang lebih baik dan responsif. Hal ini sangat penting bagi UMKM yang beroperasi di lingkungan yang memiliki konektivitas internet yang mungkin tidak selalu stabil, sehingga pemilihan teknologi yang tepat dapat meningkatkan efisiensi operasional dan kepuasan pelanggan. Untuk pengembangan aplikasi, disarankan bahwa untuk aplikasi kasir yang banyak menangani data teks, Retrofit lebih direkomendasikan. Sebaliknya, Ktor Client lebih cocok digunakan untuk aplikasi yang memproses data gambar, mengingat karakteristik dan keunggulan masing-masing *library* dalam situasi tertentu.

Rekomendasi untuk penelitian lebih lanjut mencakup pengujian terhadap *library* lain, seperti Fuel atau OkHttp API, guna mengeksplorasi potensi keunggulan mereka dalam konteks aplikasi kasir. Selain itu, pengujian aplikasi dalam kondisi jaringan ekstrem, seperti jaringan yang sangat lambat atau tidak stabil, dapat memberikan wawasan tambahan mengenai ketahanan dan kemampuan masing-masing *library* dalam situasi lebih nyata.

DAFTAR PUSTAKA

- [1] A. N. Hanum and A. Sinarasri, "Analisis Faktor-Faktor yang Mempengaruhi Adopsi E-Commerce dan Dampaknya Terhadap Kinerja UMKM (Studi Kasus UMKM di Kota Semarang)," *MAKSIMUM*, vol. 7, no. 1, p. 1, Nov. 2019, doi: 10.26714/mki.7.1.2017.1-15.
- [2] N. Marina and R. M. Pd, "Penerapan Pembukuan Digitalisasi Informasi pada UMKM Desa Sumberjaya," vol. 2, no. 2, 2022.
- [3] M. I. Alamsyahbana *et al.*, "Pencatatan Pembukuan Berbasis Digital Dengan Menggunakan Aplikasi Buku Kas (Studi Kasus Pelaku UMKM Desa Toapaya Selatan)," *Communnity Development Journal*, vol. 4, no. 6, 2023, doi: 10.31004/cdj.v4i6.23585.
- [4] W. Monalisa and B. Prabowo, "Analisis Penggunaan Aplikasi Kasir pada pelaku UMKM Kecamatan Tampan Kota Pekanbaru Riau," vol. 1, pp. 145–151, 2023, doi: 10.59581/jap-widyakarya.v1i1.619.
- [5] H. Listiyono, E. Nur Wahyudi, D. Agus Diartono, U. Stikubank Semarang Jl Tri Lomba Juang No, and M. Semarang, "Dinamika Implementasi QRIS: Meninjau Peluang dan Tantangan bagi UMKM Indonesia," vol. 8, Jul. 2024, doi: 10.37817/ikraith-informatika.v8i2.

-
- [6] H. Khatri, “Over 20% of smartphone users in Indonesia see average download speeds below 10Mbps | Opensignal.” Accessed: Jul. 14, 2024. [Online]. Available: <https://www.opensignal.com/2023/10/10/over-20-of-smartphone-users-in-indonesia-see-average-download-speeds-below-10mbps>
- [7] K. W. K. W. I, “Pengembangan Frontend Aplikasi Sistem Monitoring Detak Jantung Berbasis Mobile dengan Koneksi IoT,” *Telkom Open Library*, 2024.
- [8] M. S. Ferryansyah, M. Tri Ananta, and L. Fanani, “Analisis Performansi HTTP Networking Library pada Android (Studi Kasus: Portal Berita),” 2018. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [9] “ISO/IEC 25010:2023(en), Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model.” Accessed: Oct. 08, 2024. [Online]. Available: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:25010:ed-2:v1:en>
- [10] B. Vernanda, A. Prayoga, A. Pinandito, and A. P. Kharisma, “Studi Performa Android Networking Library antara Fast Android Network Library, Retrofit dan OkHttp,” 2023. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [11] “Android network libraries.” Accessed: Oct. 05, 2024. [Online]. Available: <https://www.appbrain.com/stats/libraries/tag/network/android-network-libraries>
- [12] “Ktor: Build Asynchronous Servers and Clients in Kotlin | Ktor Framework.” Accessed: Oct. 08, 2024. [Online]. Available: <https://ktor.io/>
- [13] “Kotlin Programming Language.” Accessed: Oct. 08, 2024. [Online]. Available: <https://kotlinlang.org/>
- [14] “TrafficStats | Android Developers.” Accessed: Sep. 29, 2024. [Online]. Available: <https://developer.android.com/reference/kotlin/android/net/TrafficStats>
- [15] “Debug.MemoryInfo | Android Developers.” Accessed: Sep. 29, 2024. [Online]. Available: <https://developer.android.com/reference/android/os/Debug.MemoryInfo>
- [16] I. M. Riyadhi, Intan Purnamasari, and Kamal Prihandani, “Penerapan Pola Arsitektur MVVM pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android,” *INFOTECH journal*, vol. 9, no. 1, pp. 147–158, May 2023, doi: 10.31949/infotech.v9i1.5246.
- [17] B. Amary, “Implementasi HTTP Library Retrofit, Volley, dan HttpURLConnection dalam Akses Data,” 2023.

