

OPTIMASI PERFORMA APLIKASI MOBILE DENGAN PENERAPAN PAGING DAN INFINITE SCROLL MENGGUNAKAN BLOC DI FLUTTER

Muhammad Hadi Royan Firdaus^{1*}, Rini Mayasari², Riza Ibnu Adam³

^{1,2,3}Universitas Singaperbangsa Karawang: Jl. HS. Ronggowaluyo Telukjambe Timur, Karawang, Jawa Barat 41361; (0267) 641177

Keywords:

Flutter;
Paging;
Infinte Sroll;
Mobile App.

Corespondent Email:

2110631170079@student.uns
ika.ac.id

Abstrak. Kualitas performa aplikasi mobile berpengaruh signifikan terhadap pengalaman pengguna, terutama dari segi kecepatan akses, manajemen memori yang efisien, dan kemampuan respons yang baik. Penelitian ini bertujuan untuk menganalisis dan membandingkan performa satu aplikasi mobile sebelum dan sesudah dioptimalkan menggunakan teknik paging dan infinite scroll dengan state management Bloc pada framework Flutter. Metode yang digunakan adalah pra-eksperimen dengan one-group pretest-posttest design, di mana aplikasi diuji pada kondisi awal (tanpa optimasi) dan kondisi setelah diterapkan teknik optimasi. Hasil pengujian menunjukkan bahwa setelah optimasi, aplikasi mengalami peningkatan performa signifikan pada waktu pemuatan data dan penggunaan memori. Hal ini disebabkan karena data dimuat secara bertahap, sehingga mengurangi beban proses. Responsivitas aplikasi tetap stabil pada kedua kondisi karena Flutter mendukung pemuatan komponen secara efisien melalui mekanisme lazy loading pada ListView. Dengan demikian, penerapan teknik paging dan infinite scroll terbukti efektif dalam meningkatkan efisiensi performa aplikasi mobile, terutama pada aplikasi berbasis Flutter yang menangani data dalam jumlah besar.



Copyright © **JITET** (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. The quality of mobile application performance has a significant impact on user experience, especially in terms of access speed, efficient memory management, and good responsiveness. This study aims to analyze and compare the performance of one mobile application before and after being optimized using paging and infinite scroll techniques with Bloc state management in the Flutter framework. The method used is a pre-experiment with a one-group pretest-posttest design, where the application is tested in the initial condition (without optimization) and the condition after applying the optimization technique. The test results showed that after optimization, the application experienced significant performance improvements in data loading time and memory usage. This is because the data is loaded incrementally, thus reducing the process load. The responsiveness of the application remains stable in both conditions because Flutter supports efficient component loading through the lazy loading mechanism on ListView. Thus, the application of paging and infinite scroll techniques has proven effective in improving the performance efficiency of mobile applications, especially in Flutter-based applications that handle large amounts of data..

1. PENDAHULUAN

Kemajuan teknologi dan informasi di era ini berkembang dengan sangat cepat. Salah satu contohnya adalah semakin banyaknya penggunaan akses internet di masyarakat untuk

memperoleh berbagai informasi. Inovasi dari teknologi ini memungkinkan akses informasi di mana saja dan kapan saja menggunakan smartphone dan komputer yang terhubung ke internet [1].

Perkembangan akses internet yang pesat telah mendorong peningkatan permintaan terhadap aplikasi mobile. Penyebaran smartphone yang semakin meluas berbanding lurus dengan pertumbuhan aplikasi mobile, yang dirancang untuk memudahkan kehidupan manusia dalam berbagai aspek. Aplikasi-aplikasi ini mencakup layanan utama seperti belanja online, jejaring sosial, dan perbankan digital, serta banyak lagi [2].

Pengguna aplikasi mobile saat ini mengharapkan pengalaman yang cepat dan responsif terutama ketika mereka berinteraksi dengan data dalam jumlah besar. Menurut beberapa penelitian, aplikasi yang menunjukkan kinerja yang buruk, seperti lambat atau tidak responsif, tidak hanya menghalangi pengguna untuk menggunakan beberapa fitur secara penuh, tetapi juga cenderung ditinggalkan oleh pengguna dalam waktu singkat, yang berdampak pada penurunan tingkat retensi pengguna dan pendapatan aplikasi, serta kemungkinan tidak bertahan lama dalam persaingan pasar [2].

Pada umumnya, aplikasi mobile memuat seluruh data sekaligus ketika pengguna mengakses halaman tertentu. Dengan cara ini walaupun sederhana, memiliki kelemahan utama seperti penggunaan memori yang tinggi dan waktu loading yang lama, terutama ketika data yang dimuat sangat besar. Akibatnya, aplikasi cenderung lambat merespons, dan interaksi pengguna, seperti scrolling, tidak terasa mulus. Selain itu, faktor lain seperti kondisi jaringan, jumlah pengguna yang bersamaan, atau penggunaan sumber daya CPU dan GPU yang berlebihan pada animasi pemuatan juga dapat memperburuk performa aplikasi [3].

Dalam penelitian Baptista (2023), implementasi teknik optimasi dengan infinite scrolling menunjukkan peningkatan signifikan dalam performa aplikasi. Waktu startup aplikasi berhasil dikurangi hingga 18,3%, sedangkan waktu pemuatan konten utama berkurang sebesar 8,5%. Selain itu, durasi frame pada scrolling menurun sehingga memberikan pengalaman pengguna yang lebih mulus dan responsif. Hal ini menunjukkan bahwa penerapan infinite scroll dapat secara signifikan meningkatkan efisiensi aplikasi mobile, memberikan pengalaman pengguna yang lebih baik, serta mengurangi beban perangkat.

Namun penelitian Baptista hanya berfokus pada optimasi dengan infinite scroll dan menggunakan Jetpack Compose pada platform Android [4]. Penelitian ini bertujuan untuk memberikan solusi dalam meningkatkan kinerja aplikasi mobile melalui penerapan teknik paging dan infinite scroll dengan menggunakan Flutter dengan cara membandingkan waktu memuat, penggunaan memori, dan responsivitas antara aplikasi versi full data load dan versi yang menggunakan paging dengan infinite scroll.

2. TINJAUAN PUSTAKA

2.1. *Aplikasi Mobile*

Aplikasi adalah perangkat lunak yang dirancang untuk digunakan langsung oleh pengguna dalam menjalankan berbagai proses di komputer. Seiring dengan perkembangan zaman, pengguna smartphone dan komputer semakin memanfaatkan aplikasi tersebut untuk memenuhi berbagai kebutuhan dalam kehidupan sehari-hari [5]. Aplikasi mobile merupakan perangkat lunak yang dikembangkan khusus untuk digunakan pada platform mobile, seperti Android atau iOS. Aplikasi ini dirancang agar kompatibel dengan perangkat mobile [6].

2.2. *Performa Aplikasi Mobile*

Performa aplikasi mobile merupakan salah satu faktor keberhasilan suatu aplikasi. Dalam konteks aplikasi mobile, performa adalah kemampuan aplikasi untuk menjalankan fungsi dan tugasnya secara efisien dan responsif. Performa aplikasi dapat dilihat dari penggunaan sumber dayanya seperti, CPU, memory, dan baterai [7].

2.3. *Paging*

Paging digunakan untuk membagi data agar sesuai dengan kebutuhan serta kapasitas pengguna. Teknik ini sangat berguna ketika jumlah data yang diterima sangat besar. Tanpa implementasi paging, aplikasi akan memuat seluruh data sekaligus, yang dapat menyebabkan penurunan performa seperti waktu muat yang lambat. Dengan penerapan paging, aplikasi dapat menangani data besar secara lebih efisien karena tidak perlu menunggu seluruh data dimuat sekaligus [8].

2.4. *Infinite Scroll*

Infinite scroll merupakan elemen antarmuka pengguna yang kini telah diintegrasikan di hampir semua platform media sosial utama serta berbagai kategori 9 aplikasi dan situs web lainnya. Fitur ini menggantikan teknik tradisional, di mana pengguna sebelumnya harus mengklik tautan atau tombol untuk memuat halaman baru. Dengan infinite scroll, pengguna cukup menggulir ke bawah, dan konten baru akan dimuat secara dinamis di bagian bawah halaman [9].

2.5. Bloc

Bloc (Business Logic Component) adalah sebuah desain pattern yang memisahkan lapisan presentasi dengan logika bisnis, sehingga memudahkan developer dalam mengonversi event menjadi state. Dengan menggunakan Bloc, developer dapat memodifikasi kode tanpa mempengaruhi kode lainnya, membuat kode mudah dipahami [10].

2.6. Flutter

Flutter adalah framework yang dibangun berdasarkan SDK (Software Development Kit). Framework yang bersifat open-source ini dikembangkan oleh Google dan dapat berjalan di berbagai sistem operasi seperti Android, iOS, Windows, Linux, dan MacOS hanya dengan menggunakan satu basis kode yang ditulis dalam bahasa Dart. Selain kemampuannya untuk menghasilkan aplikasi multi-platform, Flutter juga memiliki beberapa keunggulan lain, seperti proses rendering yang responsif, antarmuka pengguna yang dapat dibangun melalui widget-widget yang fleksibel dan dapat disesuaikan berdasarkan konfigurasi serta state aplikasi. Hal ini membuat Flutter sangat mendukung integrasi dengan berbagai library visualisasi data, sehingga sangat ideal untuk pengembangan aplikasi yang memerlukan sistem skoring dan dashboard visualisasi data[11].

2.7. Python

Python, sebagai salah satu bahasa pemrograman yang dikembangkan oleh Guido van Rossum pada tahun 1990, telah menjadi salah satu yang paling populer dan banyak digunakan di berbagai sektor termasuk dalam analisis data. Dengan berbagai library seperti NumPy, SciPy, Pandas, dan Matplotlib, python

mempermudah pengolahan, manipulasi, analisis, dan visualisasi data secara efisien [12].

2.8. Penelitian Eksperimen

Salah satu metode pada penelitian kuantitatif adalah metode eksperimen, metode eksperimen digunakan untuk mengetahui hubungan sebab-akibat dengan membandingkan satu atau lebih kelompok eksperimen yang menerima perlakuan (treatment) dengan satu atau lebih kelompok eksperimen yang tidak menerima perlakuan [13].

Design dalam metode eksperimen dibagi menjadi 3 jenis yaitu eksperimen murni (true experimental design), quasi eksperimen (quasy experimental design), dan pra-eksperimen (pre-experimental design). Dalam desain penelitian pra-eksperimen tidak ada kelompok kontrol, sehingga hanya terdapat satu kelompok intervensi, serta pada desain ini tidak ada randomisasi sampel. Dalam desain ini kelompok akan melakukan tes awal (pretest) kemudian kelompok akan diberikan perlakuan (treatment), setelah itu kelompok tersebut akan melakukan tes akhir (posttest). Desain ini juga disebut one-group pretest-posttest design [14].

2.9. Analisis Data Statistik

Analisis merupakan suatu proses sistematis yang mencakup pengumpulan dan pengolahan data yang diperoleh melalui wawancara, catatan lapangan, serta dokumentasi. Tujuan dari analisis ini adalah untuk menjawab pertanyaan penelitian atau menguji hipotesis yang telah diajukan dengan menyajikan data secara sistematis [15].

Saat ini, perkembangan teknologi telah menghilangkan kebutuhan untuk melakukan analisis statistik secara manual dengan hadirnya berbagai perangkat lunak. Beragam aplikasi telah dikembangkan untuk keperluan analisis statistik, di antaranya adalah Statistical Package for the Social Sciences (SPSS), Analysis of Moment Structures (AMOS), Microstat, dan EViews. Penggunaan perangkat lunak ini memberikan kemudahan serta meningkatkan efisiensi bagi para peneliti dalam melakukan analisis data [15].

Salah satu metode statistik yang digunakan dalam analisis data adalah metode non-parametrik, yang juga dikenal sebagai metode distribusi bebas karena tidak

memerlukan asumsi bahwa data berdistribusi normal. Metode ini dapat diterapkan pada data berskala nominal maupun ordinal [16]. Salah satu contoh dari metode non-parametrik adalah Uji Wilcoxon Signed-Rank.

2.9.1. Uji Normalitas

Pengujian normalitas bertujuan untuk mengevaluasi apakah suatu data berasal dari populasi dengan distribusi normal atau tidak. Salah satu metode yang sering digunakan dalam pengujian ini adalah uji Shapiro-Wilk [15].

2.9.2 Uji Homogenitas

Uji homogenitas bertujuan untuk mengevaluasi apakah beberapa kumpulan data dari suatu populasi memiliki varians yang seragam atau tidak. Uji homogenitas yang sering digunakan adalah Levene [15].

2.9.3 Wilcoxon Signed-Rank Test

Uji Wilcoxon Sign Rank adalah metode non-parametrik yang umum digunakan untuk data kualitatif (dengan skala nominal atau ordinal), maupun data kuantitatif yang tidak terdistribusi normal. Metode ini digunakan untuk menguji hipotesis komparatif atau membandingkan dua sampel yang berpasangan atau berkorelasi untuk mengetahui apakah ada perbedaan di antara keduanya [17].

3. METODE PENELITIAN

Metode penelitian ini menggunakan metode eksperimen dengan jenis praeeksperimen dengan bentuk one-group pretest-posttest design. Dengan desain penelitian tersebut maka akan dilakukan pengujian awal (pretest) sebelum dioptimasi dan pengujian akhir (posttest). Hasil pengujian akan dianalisis untuk mengetahui pengaruh teknik paging dan infinite scroll terhadap performa aplikasi.

3.1. Objek Penelitian

Objek pada penelitian optimasi performa aplikasi mobile dengan penerapan paging dan infinite scroll menggunakan bloc di Flutter ini adalah aplikasi mobile yang diuji dengan metode pra-eksperimen. Aplikasi ini akan diuji sebelum dioptimasi dan sesudah dioptimasi menggunakan paging dan infinite scroll.

3.3. Variabel Penelitian

Variabel penelitian ini terdiri dari variabel terikat (dependen) dan variabel bebas (Independen).

Variabel terikat merupakan variabel yang dipengaruhi atau mengalami perubahan sebagai dampak dari pengaruh variabel bebas. Variabel terikat pada penelitian ini yaitu performa aplikasi dengan memperhatikan waktu memuat, penggunaan memori, dan responsivitas.

Sedangkan, Variabel bebas adalah variabel yang memengaruhi atau menyebabkan perubahan pada variabel terikat. Variabel bebas pada penelitian ini yaitu teknik optimasi yang terdiri dari paging dan infinite scroll.

3.4. Desain Penelitian

Penelitian ini menerapkan desain one-group pretest-posttest, di mana hanya terdapat satu kelompok tanpa kelompok kontrol dan tanpa proses randomisasi sampel.

O1	X	O1
----	---	----

Keterangan:

O1: Tes awal sebelum dioptimasi (pretest)

X: Perlakuan berupa optimasi (treatment)

O2: Tes akhir setelah dioptimasi (posttest)

Perlakuan yang diberikan adalah optimasi aplikasi menggunakan teknik paging dan infinite scroll dan pengukuran dilakukan pada aplikasi mobile dengan memperhatikan waktu memuat, penggunaan memori dan responsivitas.

3.5. Analisis Data

Data dalam penelitian ini dikumpulkan melalui pengujian yang dilakukan menggunakan emulator Android. Data performa aplikasi, seperti waktu memuat, penggunaan memori, dan responsivitas, dikumpulkan menggunakan alat profiling, seperti Flutter DevTools. Teknik analisis data dalam penelitian ini dilakukan melalui beberapa tahapan sebagai berikut:

3.5.1 Uji Normalitas

Dalam penelitian ini, pengujian dilakukan menggunakan metode Shapiro-Wilk dengan bantuan program python, dengan ketentuan:

- A. Jika nilai signifikansi (*Sig*) > 0,05, maka data berdistribusi normal.

- B. Jika nilai signifikansi (Sig) < 0,05, maka data tidak berdistribusi normal.

Hipotesis untuk setiap varibel:

1. Waktu Memuat:

- H_0 : Data waktu memuat berdistribusi normal.
- H_1 : Data waktu memuat tidak berdistribusi normal.

2. Penggunaan Memori:

- H_0 : Data penggunaan memori berdistribusi normal.
- H_1 : Data penggunaan memori tidak berdistribusi normal.

3. Responsivitas:

- H_0 : Data responsivitas berdistribusi normal.
- H_1 : Data responsivitas tidak berdistribusi normal.

3.5.2 Uji Homogenitas

Pada penelitian ini uji homogenitas menggunakan uji Levene dengan bantuan program python, dengan ketentuan:

- A. Jika nilai signifikansi (Sig) > 0,05, maka data disebut homogen.
- B. Jika nilai signifikansi (Sig) < 0,05, maka data tidak homogen.

Hipotesis untuk setiap varibel:

1. Waktu Memuat:

- H_0 : Data waktu memuat sebelum dan sesudah optimasi adalah homogen.
- H_1 : Data waktu memuat sebelum dan sesudah optimasi tidak homogen.

2. Penggunaan Memori:

- H_0 : Data penggunaan memori sebelum dan sesudah optimasi adalah homogen.
- H_1 : Data penggunaan memori sebelum dan sesudah optimasi tidak homogen. normal.

3. Responsivitas:

- H_0 : Data responsivitas sebelum dan sesudah optimasi adalah homogen.
- H_1 : Data responsivitas sebelum dan sesudah optimasi tidak homogen.

3.5.3 Uji Hipotesis

Uji hipotesis menggunakan wilcoxon signed-rank test yang digunakan untuk menguji apakah terdapat perbedaan signifikan antara performa aplikasi sebelum dan sesudah optimasi. Data yang dianalisis meliputi waktu memuat aplikasi, penggunaan memori, dan responsivitas (FPS). Pengujian dilakukan menggunakan python untuk membandingkan dua kondisi: sebelum optimasi (tanpa paging dan infinite scroll) dan sesudah optimasi (dengan paging dan infinite scroll), dengan ketentuan ketentuan:

- A. Jika nilai signifikansi (Sig) > 0,05, maka H_0 diterima.
- B. Jika nilai signifikansi (Sig) < 0,05, maka H_0 ditolak.

Uji hipotesis dilakukan untuk masing-masing variabel sebagai berikut:

1. Waktu Memuat

- H_0 : Tidak terdapat perbedaan signifikan dalam waktu memuat sebelum dan sesudah optimasi menggunakan teknik paging dan infinite scroll.
- H_1 : Terdapat perbedaan signifikan dalam waktu memuat sebelum dan sesudah optimasi menggunakan teknik paging dan infinite scroll.

2. Penggunaan Memori

- H_0 : Tidak terdapat perbedaan signifikan dalam penggunaan memori sebelum dan sesudah optimasi menggunakan teknik paging dan infinite scroll
- H_1 : Terdapat perbedaan signifikan dalam penggunaan memori sebelum dan sesudah optimasi menggunakan teknik paging dan infinite scroll.

3. Responsivitas

- H_0 : Tidak terdapat perbedaan signifikan dalam responsivitas (FPS) sebelum dan sesudah optimasi menggunakan teknik paging dan infinite scroll.
- H_1 : Terdapat perbedaan signifikan dalam responsivitas (FPS) sebelum

dan sesudah optimasi menggunakan teknik paging dan infinite scroll

3.6. Rancangan Penelitian

Rancangan penelitian ini terdiri dari 4 tahap persiapan, pelaksanaan, analisis data, dan penyelesaian.

3.6.1. Persiapan

Pada tahap persiapan, peneliti terlebih dahulu melakukan studi literatur dengan mengumpulkan berbagai referensi yang relevan terkait optimasi performa aplikasi mobile, khususnya mengenai penerapan paging dan infinite scroll menggunakan state management Bloc pada Flutter. Selain itu, peneliti menyiapkan perangkat keras berupa laptop dengan spesifikasi memadai serta software pendukung seperti Flutter SDK, Visual Studio Code, dan Python. Pada tahap ini peneliti juga menyusun rancangan aplikasi versi awal, yaitu aplikasi yang dibangun menggunakan Bloc tetapi belum menerapkan optimasi berupa paging dan infinite scroll. Selain itu, menyiapkan aplikasi awal tanpa optimasi paging dan infinite scroll serta program Python untuk analisis data yang diperoleh dari hasil pengujian performa aplikasi.

3.6.2. Pelaksanaan

Tahap pelaksanaan dimulai dengan melakukan pengujian aplikasi sebelum dioptimasi (pretest) untuk memperoleh data awal performa. Pada tahap ini dilakukan pengukuran terhadap indikator performa, yaitu penggunaan memori, waktu muat aplikasi, dan responsivitas (FPS). Setelah itu, peneliti menerapkan teknik optimasi dengan mengintegrasikan teknik paging, dan infinite scroll pada aplikasi, lalu melakukan pengujian kembali (posttest) pada aplikasi yang telah dioptimasi. Pengukuran performa dilakukan menggunakan Flutter DevTools untuk memantau penggunaan memori serta responsivitas aplikasi, sedangkan waktu muat aplikasi diukur menggunakan stopwatch yang dikombinasikan dengan log aplikasi.

3.6.3. Analisis Data

Data penelitian diperoleh dari hasil pengujian pretest dan posttest yang dilakukan pada aplikasi sebelum dan sesudah penerapan teknik optimasi. Parameter yang diukur

meliputi waktu memuat aplikasi, penggunaan memori, serta responsivitas aplikasi dalam bentuk frame per second (FPS). Pengambilan data dilakukan dengan memanfaatkan profiling tools Flutter DevTools agar hasil yang diperoleh lebih akurat dan terukur.

Tahap analisis data diawali dengan pengujian normalitas menggunakan metode Shapiro-Wilk untuk mengetahui apakah data berdistribusi normal. Selanjutnya dilakukan uji homogenitas menggunakan metode Levene sebagai syarat lanjutan sebelum uji hipotesis. Uji hipotesis yang digunakan adalah uji Wilcoxon SignedRank, yang sesuai digunakan untuk data berpasangan dengan kondisi pretest dan posttest. Seluruh proses analisis data dilakukan dengan bantuan bahasa pemrograman Python.

3.6.4. Penyelesaian

Pada tahap ini, hasil analisis data diinterpretasikan untuk menjawab rumusan masalah penelitian. Interpretasi dilakukan dengan membandingkan hasil pretest dan posttest sehingga dapat diketahui pengaruh penerapan teknik paging dan infinite scroll terhadap performa aplikasi. Hasil pengujian kemudian menjadi dasar dalam menarik kesimpulan mengenai efektivitas optimasi yang dilakukan, serta memberikan gambaran apakah metode tersebut dapat dijadikan pendekatan yang tepat untuk meningkatkan kinerja aplikasi mobile berbasis Flutter.

4. HASIL DAN PEMBAHASAN

Hasil penelitian ini bertujuan untuk menganalisis dan membandingkan performa aplikasi mobile sebelum dan sesudah dilakukan optimasi. Aplikasi yang digunakan adalah aplikasi berbasis Flutter.

Optimasi dilakukan dengan menerapkan teknik paging dan infinite scroll, yang bertujuan untuk meningkatkan efisiensi performa dari sisi waktu memuat, penggunaan memori, dan responsivitas (FPS). Aplikasi diuji sebanyak 15 kali sebelum optimasi (pretest) dan 15 kali setelah optimasi (posttest).

Pengukuran dilakukan menggunakan alat bantu seperti Flutter DevTools untuk memperoleh data konsumsi memori (MB) dan frame rate (FPS). Sementara itu, untuk mengukur waktu memuat digunakan kode stopwatch pada aplikasi. Stopwatch dimulai

saat proses pemanggilan data, dan dihentikan setelah seluruh komponen (card) selesai dirender ke layar. Nilai waktu yang tercatat kemudian dicetak ke terminal dalam satuan mili detik.

4.1. Persiapan

Pada tahap persiapan, penelitian ini dilakukan dengan memanfaatkan perangkat keras berupa sebuah laptop dengan spesifikasi memadai agar mampu menjalankan proses pengembangan dan pengujian aplikasi dengan lancar. Perangkat lunak yang digunakan meliputi Flutter SDK sebagai framework utama pengembangan aplikasi, Visual Studio Code sebagai editor kode, serta Python yang dipakai untuk melakukan analisis data hasil pengujian performa. Selain itu, beberapa library pendukung turut disiapkan, salah satunya adalah flutter_bloc untuk manajemen state, serta library http untuk kebutuhan pengambilan data dari API.

Aplikasi awal yang dikembangkan pada tahap ini dibangun menggunakan Flutter dan Bloc sebagai state management, namun belum dilengkapi dengan mekanisme optimasi seperti paging dan infinite scroll. Sehingga seluruh data ditampilkan sekaligus pada satu waktu.

```
skripsi_app> lib> src> features> users> presentation> bloc> user_bloc.dart > UserBloc
1 import 'package:bloc/bloc.dart';
2 import 'package:equatable/equatable.dart';
3 import 'package:skripsi_app/src/features/users/domain/entities/user.dart';
4 import 'package:skripsi_app/src/features/users/domain/usecase/get_users.dart';
5
6 part 'user_event.dart';
7 part 'user_state.dart';
8
9 class UserBloc extends Bloc<UserEvent, UserState> {
10   final GetUsers getusers;
11
12   UserBloc({required this.getUsers}) : super(UserInitial()) {
13     onUserEvent<(event, emit)>();
14     onFetchUserEvent<(onFetchUsers)>();
15   }
16
17   Future<void> _onFetchUsers(
18     FetchUserEvent event,
19     Emitter<UserState> emit,
20   ) async {
21     emit(UserLoadingState());
22     try {
23       final users = await getusers(event.size);
24       print('UserBloc: total users: ${users.length}'); // Don't invoke 'print' in production code.
25       emit(UserLoadedState(users: users));
26     } catch (e) {
27       print('UserBloc Error: $e'); // Don't invoke 'print' in production code.
28       emit(UserErrorState(message: e.toString()));
29     }
30   }
31 }
32
```

Gambar 1. Implementasi Bloc tanpa optimasi

Kode di atas adalah implementasi kelas UserBloc yang digunakan untuk mengatur logika pengambilan data pengguna.

Selanjutnya, pada sisi tampilan, Ketika data telah dimuat, ListView.builder akan menampilkan seluruh data pengguna dalam bentuk daftar (card) secara penuh. Pada tahap ini, belum diterapkan teknik optimasi berupa

paging maupun infinite scroll, sehingga seluruh data akan dimuat sekaligus.

```
skripsi_app> lib> src> features> users> presentation> pages> home_page.dart > HomePageState > build
1 class _HomePageState extends StatelessWidget {
2   @override
3   Widget build(BuildContext context) {
4     final UserBloc userBloc = context.read<UserBloc>();
5     final UserState state = userBloc.state;
6     if (state is UserInitial) {
7       return Center(child: CircularProgressIndicator());
8     } else if (state is UserLoaded) {
9       final List<User> users = state.users;
10      final int itemCount = users.length;
11      final double itemWidth = 150.0;
12      final double itemHeight = 150.0;
13      final double itemMargin = 10.0;
14      final double itemRadius = 10.0;
15      final double itemElevation = 5.0;
16      final double itemShadowColor = Colors.black;
17      final double itemShadowOffsetX = 5.0;
18      final double itemShadowOffsetY = 5.0;
19      final double itemShadowBlurRadius = 10.0;
20      final double itemShadowSpreadRadius = 5.0;
21      final double itemShadowRadius = 10.0;
22      final double itemShadowColor2 = Colors.black;
23      final double itemShadowOffsetX2 = 5.0;
24      final double itemShadowOffsetY2 = 5.0;
25      final double itemShadowBlurRadius2 = 10.0;
26      final double itemShadowSpreadRadius2 = 5.0;
27      final double itemShadowRadius2 = 10.0;
28      final double itemRadius2 = 10.0;
29      final double itemElevation2 = 5.0;
30      final double itemWidth2 = 150.0;
31      final double itemHeight2 = 150.0;
32      final double itemMargin2 = 10.0;
33      final double itemRadius3 = 10.0;
34      final double itemElevation3 = 5.0;
35      final double itemWidth3 = 150.0;
36      final double itemHeight3 = 150.0;
37      final double itemMargin3 = 10.0;
38      final double itemRadius4 = 10.0;
39      final double itemElevation4 = 5.0;
40      final double itemWidth4 = 150.0;
41      final double itemHeight4 = 150.0;
42      final double itemMargin4 = 10.0;
43      final double itemRadius5 = 10.0;
44      final double itemElevation5 = 5.0;
45      final double itemWidth5 = 150.0;
46      final double itemHeight5 = 150.0;
47      final double itemMargin5 = 10.0;
48      final double itemRadius6 = 10.0;
49      final double itemElevation6 = 5.0;
50      final double itemWidth6 = 150.0;
51      final double itemHeight6 = 150.0;
52      final double itemMargin6 = 10.0;
53      final double itemRadius7 = 10.0;
54      final double itemElevation7 = 5.0;
55      final double itemWidth7 = 150.0;
56      final double itemHeight7 = 150.0;
57      final double itemMargin7 = 10.0;
58      final double itemRadius8 = 10.0;
59      final double itemElevation8 = 5.0;
60      final double itemWidth8 = 150.0;
61      final double itemHeight8 = 150.0;
62      final double itemMargin8 = 10.0;
63      final double itemRadius9 = 10.0;
64      final double itemElevation9 = 5.0;
65      final double itemWidth9 = 150.0;
66      final double itemHeight9 = 150.0;
67      final double itemMargin9 = 10.0;
68      final double itemRadius10 = 10.0;
69      final double itemElevation10 = 5.0;
70      final double itemWidth10 = 150.0;
71      final double itemHeight10 = 150.0;
72      final double itemMargin10 = 10.0;
73      final double itemRadius11 = 10.0;
74      final double itemElevation11 = 5.0;
75      final double itemWidth11 = 150.0;
76      final double itemHeight11 = 150.0;
77      final double itemMargin11 = 10.0;
78      final double itemRadius12 = 10.0;
79      final double itemElevation12 = 5.0;
80      final double itemWidth12 = 150.0;
81      final double itemHeight12 = 150.0;
82      final double itemMargin12 = 10.0;
83      final double itemRadius13 = 10.0;
84      final double itemElevation13 = 5.0;
85      final double itemWidth14 = 150.0;
86      final double itemHeight14 = 150.0;
87      final double itemMargin14 = 10.0;
88      final double itemRadius15 = 10.0;
89      final double itemElevation15 = 5.0;
90      final double itemWidth16 = 150.0;
91      final double itemHeight16 = 150.0;
92      final double itemMargin16 = 10.0;
93      final double itemRadius17 = 10.0;
94      final double itemElevation17 = 5.0;
95      final double itemWidth18 = 150.0;
96      final double itemHeight18 = 150.0;
97      final double itemMargin18 = 10.0;
98      final double itemRadius19 = 10.0;
99      final double itemElevation19 = 5.0;
100     final double itemWidth20 = 150.0;
101     final double itemHeight20 = 150.0;
102     final double itemMargin20 = 10.0;
103     final double itemRadius21 = 10.0;
104     final double itemElevation21 = 5.0;
105     final double itemWidth22 = 150.0;
106     final double itemHeight22 = 150.0;
107     final double itemMargin22 = 10.0;
108     final double itemRadius23 = 10.0;
109     final double itemElevation23 = 5.0;
110     final double itemWidth24 = 150.0;
111     final double itemHeight24 = 150.0;
112     final double itemMargin24 = 10.0;
113     final double itemRadius25 = 10.0;
114     final double itemElevation25 = 5.0;
115     final double itemWidth26 = 150.0;
116     final double itemHeight26 = 150.0;
117     final double itemMargin26 = 10.0;
118     final double itemRadius27 = 10.0;
119     final double itemElevation27 = 5.0;
120     final double itemWidth28 = 150.0;
121     final double itemHeight28 = 150.0;
122     final double itemMargin28 = 10.0;
123     final double itemRadius29 = 10.0;
124     final double itemElevation29 = 5.0;
125     final double itemWidth30 = 150.0;
126     final double itemHeight30 = 150.0;
127     final double itemMargin30 = 10.0;
128     final double itemRadius31 = 10.0;
129     final double itemElevation31 = 5.0;
130     final double itemWidth32 = 150.0;
131     final double itemHeight32 = 150.0;
132     final double itemMargin32 = 10.0;
133     final double itemRadius33 = 10.0;
134     final double itemElevation34 = 5.0;
135     final double itemWidth35 = 150.0;
136     final double itemHeight35 = 150.0;
137     final double itemMargin35 = 10.0;
138     final double itemRadius36 = 10.0;
139     final double itemElevation37 = 5.0;
140     final double itemWidth38 = 150.0;
141     final double itemHeight38 = 150.0;
142     final double itemMargin38 = 10.0;
143     final double itemRadius39 = 10.0;
144     final double itemElevation40 = 5.0;
145     final double itemWidth41 = 150.0;
146     final double itemHeight41 = 150.0;
147     final double itemMargin41 = 10.0;
148     final double itemRadius42 = 10.0;
149     final double itemElevation43 = 5.0;
150     final double itemWidth44 = 150.0;
151     final double itemHeight44 = 150.0;
152     final double itemMargin44 = 10.0;
153     final double itemRadius45 = 10.0;
154     final double itemElevation46 = 5.0;
155     final double itemWidth47 = 150.0;
156     final double itemHeight47 = 150.0;
157     final double itemMargin47 = 10.0;
158     final double itemRadius48 = 10.0;
159     final double itemElevation49 = 5.0;
160     final double itemWidth50 = 150.0;
161     final double itemHeight50 = 150.0;
162     final double itemMargin50 = 10.0;
163     final double itemRadius51 = 10.0;
164     final double itemElevation52 = 5.0;
165     final double itemWidth53 = 150.0;
166     final double itemHeight53 = 150.0;
167     final double itemMargin53 = 10.0;
168     final double itemRadius54 = 10.0;
169     final double itemElevation55 = 5.0;
170     final double itemWidth56 = 150.0;
171     final double itemHeight56 = 150.0;
172     final double itemMargin56 = 10.0;
173     final double itemRadius57 = 10.0;
174     final double itemElevation58 = 5.0;
175     final double itemWidth59 = 150.0;
176     final double itemHeight59 = 150.0;
177     final double itemMargin59 = 10.0;
178     final double itemRadius60 = 10.0;
179     final double itemElevation61 = 5.0;
180     final double itemWidth62 = 150.0;
181     final double itemHeight62 = 150.0;
182     final double itemMargin62 = 10.0;
183     final double itemRadius63 = 10.0;
184     final double itemElevation64 = 5.0;
185     final double itemWidth65 = 150.0;
186     final double itemHeight65 = 150.0;
187     final double itemMargin65 = 10.0;
188     final double itemRadius66 = 10.0;
189     final double itemElevation67 = 5.0;
190     final double itemWidth68 = 150.0;
191     final double itemHeight68 = 150.0;
192     final double itemMargin68 = 10.0;
193     final double itemRadius69 = 10.0;
194     final double itemElevation70 = 5.0;
195     final double itemWidth71 = 150.0;
196     final double itemHeight71 = 150.0;
197     final double itemMargin71 = 10.0;
198     final double itemRadius72 = 10.0;
199     final double itemElevation73 = 5.0;
200     final double itemWidth74 = 150.0;
201     final double itemHeight74 = 150.0;
202     final double itemMargin74 = 10.0;
203     final double itemRadius75 = 10.0;
204     final double itemElevation76 = 5.0;
205     final double itemWidth77 = 150.0;
206     final double itemHeight77 = 150.0;
207     final double itemMargin77 = 10.0;
208     final double itemRadius78 = 10.0;
209     final double itemElevation79 = 5.0;
210     final double itemWidth80 = 150.0;
211     final double itemHeight80 = 150.0;
212     final double itemMargin80 = 10.0;
213     final double itemRadius81 = 10.0;
214     final double itemElevation82 = 5.0;
215     final double itemWidth83 = 150.0;
216     final double itemHeight83 = 150.0;
217     final double itemMargin83 = 10.0;
218     final double itemRadius84 = 10.0;
219     final double itemElevation85 = 5.0;
220     final double itemWidth86 = 150.0;
221     final double itemHeight86 = 150.0;
222     final double itemMargin86 = 10.0;
223     final double itemRadius87 = 10.0;
224     final double itemElevation88 = 5.0;
225     final double itemWidth89 = 150.0;
226     final double itemHeight89 = 150.0;
227     final double itemMargin89 = 10.0;
228     final double itemRadius90 = 10.0;
229     final double itemElevation91 = 5.0;
230     final double itemWidth92 = 150.0;
231     final double itemHeight92 = 150.0;
232     final double itemMargin92 = 10.0;
233     final double itemRadius93 = 10.0;
234     final double itemElevation94 = 5.0;
235     final double itemWidth95 = 150.0;
236     final double itemHeight95 = 150.0;
237     final double itemMargin95 = 10.0;
238     final double itemRadius96 = 10.0;
239     final double itemElevation97 = 5.0;
240     final double itemWidth98 = 150.0;
241     final double itemHeight98 = 150.0;
242     final double itemMargin98 = 10.0;
243     final double itemRadius99 = 10.0;
244     final double itemElevation100 = 5.0;
245     final double itemWidth101 = 150.0;
246     final double itemHeight101 = 150.0;
247     final double itemMargin101 = 10.0;
248     final double itemRadius102 = 10.0;
249     final double itemElevation103 = 5.0;
250     final double itemWidth104 = 150.0;
251     final double itemHeight104 = 150.0;
252     final double itemMargin104 = 10.0;
253     final double itemRadius105 = 10.0;
254     final double itemElevation106 = 5.0;
255     final double itemWidth107 = 150.0;
256     final double itemHeight107 = 150.0;
257     final double itemMargin107 = 10.0;
258     final double itemRadius108 = 10.0;
259     final double itemElevation109 = 5.0;
260     final double itemWidth110 = 150.0;
261     final double itemHeight110 = 150.0;
262     final double itemMargin110 = 10.0;
263     final double itemRadius111 = 10.0;
264     final double itemElevation112 = 5.0;
265     final double itemWidth113 = 150.0;
266     final double itemHeight113 = 150.0;
267     final double itemMargin113 = 10.0;
268     final double itemRadius114 = 10.0;
269     final double itemElevation115 = 5.0;
270     final double itemWidth116 = 150.0;
271     final double itemHeight116 = 150.0;
272     final double itemMargin116 = 10.0;
273     final double itemRadius117 = 10.0;
274     final double itemElevation118 = 5.0;
275     final double itemWidth119 = 150.0;
276     final double itemHeight119 = 150.0;
277     final double itemMargin119 = 10.0;
278     final double itemRadius120 = 10.0;
279     final double itemElevation121 = 5.0;
280     final double itemWidth122 = 150.0;
281     final double itemHeight122 = 150.0;
282     final double itemMargin122 = 10.0;
283     final double itemRadius123 = 10.0;
284     final double itemElevation124 = 5.0;
285     final double itemWidth125 = 150.0;
286     final double itemHeight125 = 150.0;
287     final double itemMargin125 = 10.0;
288     final double itemRadius126 = 10.0;
289     final double itemElevation127 = 5.0;
290     final double itemWidth128 = 150.0;
291     final double itemHeight128 = 150.0;
292     final double itemMargin128 = 10.0;
293     final double itemRadius129 = 10.0;
294     final double itemElevation130 = 5.0;
295     final double itemWidth131 = 150.0;
296     final double itemHeight131 = 150.0;
297     final double itemMargin131 = 10.0;
298     final double itemRadius132 = 10.0;
299     final double itemElevation133 = 5.0;
300     final double itemWidth134 = 150.0;
301     final double itemHeight134 = 150.0;
302     final double itemMargin134 = 10.0;
303     final double itemRadius135 = 10.0;
304     final double itemElevation136 = 5.0;
305     final double itemWidth137 = 150.0;
306     final double itemHeight137 = 150.0;
307     final double itemMargin137 = 10.0;
308     final double itemRadius138 = 10.0;
309     final double itemElevation139 = 5.0;
310     final double itemWidth140 = 150.0;
311     final double itemHeight140 = 150.0;
312     final double itemMargin140 = 10.0;
313     final double itemRadius141 = 10.0;
314     final double itemElevation142 = 5.0;
315     final double itemWidth143 = 150.0;
316     final double itemHeight143 = 150.0;
317     final double itemMargin143 = 10.0;
318     final double itemRadius144 = 10.0;
319     final double itemElevation145 = 5.0;
320     final double itemWidth146 = 150.0;
321     final double itemHeight146 = 150.0;
322     final double itemMargin146 = 10.0;
323     final double itemRadius147 = 10.0;
324     final double itemElevation148 = 5.0;
325     final double itemWidth149 = 150.0;
326     final double itemHeight149 = 150.0;
327     final double itemMargin149 = 10.0;
328     final double itemRadius150 = 10.0;
329     final double itemElevation151 = 5.0;
330     final double itemWidth152 = 150.0;
331     final double itemHeight152 = 150.0;
332     final double itemMargin152 = 10.0;
333     final double itemRadius153 = 10.0;
334     final double itemElevation154 = 5.0;
335     final double itemWidth155 = 150.0;
336     final double itemHeight155 = 150.0;
337     final double itemMargin155 = 10.0;
338     final double itemRadius156 = 10.0;
339     final double itemElevation157 = 5.0;
340     final double itemWidth158 = 150.0;
341     final double itemHeight158 = 150.0;
342     final double itemMargin158 = 10.0;
343     final double itemRadius159 = 10.0;
344     final double itemElevation160 = 5.0;
345     final double itemWidth161 = 150.0;
346     final double itemHeight161 = 150.0;
347     final double itemMargin161 = 10.0;
348     final double itemRadius162 = 10.0;
349     final double itemElevation163 = 5.0;
350     final double itemWidth164 = 150.0;
351     final double itemHeight164 = 150.0;
352     final double itemMargin164 = 10.0;
353     final double itemRadius165 = 10.0;
354     final double itemElevation166 = 5.0;
355     final double itemWidth167 = 150.0;
356     final double itemHeight167 = 150.0;
357     final double itemMargin167 = 10.0;
358     final double itemRadius168 = 10.0;
359     final double itemElevation169 = 5.0;
360     final double itemWidth170 = 150.0;
361     final double itemHeight170 = 150.0;
362     final double itemMargin170 = 10.0;
363     final double itemRadius171 = 10.0;
364     final double itemElevation172 = 5.0;
365     final double itemWidth173 = 150.0;
366     final double itemHeight173 = 150.0;
367     final double itemMargin173 = 10.0;
368     final double itemRadius174 = 10.0;
369     final double itemElevation175 = 5.0;
370     final double itemWidth176 = 150.0;
371     final double itemHeight176 = 150.0;
372     final double itemMargin176 = 10.0;
373     final double itemRadius177 = 10.0;
374     final double itemElevation178 = 5.0;
375     final double itemWidth179 = 150.0;
376     final double itemHeight179 = 150.0;
377     final double itemMargin179 = 10.0;
378     final double itemRadius180 = 10.0;
379     final double itemElevation181 = 5.0;
380     final double itemWidth182 = 150.0;
381     final double itemHeight182 = 150.0;
382     final double itemMargin182 = 10.0;
383     final double itemRadius183 = 10.0;
384     final double itemElevation184 = 5.0;
385     final double itemWidth185 = 150.0;
386     final double itemHeight185 = 150.0;
387     final double itemMargin185 = 10.0;
388     final double itemRadius186 = 10.0;
389     final double itemElevation187 = 5.0;
390     final double itemWidth188 = 150.0;
391     final double itemHeight188 = 150.0;
392     final double itemMargin188 = 10.0;
393     final double itemRadius189 = 10.0;
394     final double itemElevation190 = 5.0;
395     final double itemWidth191 = 150.0;
396     final double itemHeight191 = 150.0;
397     final double itemMargin191 = 10.0;
398     final double itemRadius192 = 10.0;
399     final double itemElevation193 = 5.0;
400     final double itemWidth194 = 150.0;
401     final double itemHeight194 = 150.0;
402     final double itemMargin194 = 10.0;
403     final double itemRadius195 = 10.0;
404     final double itemElevation196 = 5.0;
405     final double itemWidth197 = 150.0;
406     final double itemHeight197 = 150.0;
407     final double itemMargin197 = 10.0;
408     final double itemRadius198 = 10.0;
409     final double itemElevation199 = 5.0;
410     final double itemWidth200 = 150.0;
411     final double itemHeight200 = 150.0;
412     final double itemMargin200 = 10.0;
413     final double itemRadius201 = 10.0;
414     final double itemElevation202 = 5.0;
415     final double itemWidth203 = 150.0;
416     final double itemHeight203 = 150.0;
417     final double itemMargin203 = 10.0;
418     final double itemRadius204 = 10.0;
419     final double itemElevation205 = 5.0;
420     final double itemWidth206 = 150.0;
421     final double itemHeight206 = 150.0;
422     final double itemMargin206 = 10.0;
423     final double itemRadius207 = 10.0;
424     final double itemElevation208 = 5.0;
425     final double itemWidth209 = 150.0;
426     final double itemHeight209 = 150.0;
427     final double itemMargin209 = 10.0;
428     final double itemRadius210 = 10.0;
429     final double itemElevation211 = 5.0;
430     final double itemWidth212 = 150.0;
431     final double itemHeight212 = 150.0;
432     final double itemMargin212 = 10.0;
433     final double itemRadius213 = 10.0;
434     final double itemElevation214 = 5.0;
435     final double itemWidth215 = 150.0;
436     final double itemHeight215 = 150.0;
437     final double itemMargin215 = 10.0;
438     final double itemRadius216 = 10.0;
439     final double itemElevation217 = 5.0;
440     final double itemWidth218 = 150.0;
441     final double itemHeight218 = 150.0;
442     final double itemMargin218 = 10.0;
443     final double itemRadius219 = 10.0;
444     final double itemElevation220 = 5.0;
445     final double itemWidth221 = 150.0;
446     final double itemHeight221 = 150.0;
447     final double itemMargin221 = 10.0;
448     final double itemRadius222 = 10.0;
449     final double itemElevation223 = 5.0;
450     final double itemWidth224 = 150.0;
451     final double itemHeight224 = 150.0;
452     final double itemMargin224 = 10.0;
453     final double itemRadius225 = 10.0;
454     final double itemElevation226 = 5.0;
455     final double itemWidth227 = 150.0;
456     final double itemHeight227 = 150.0;
457     final double itemMargin227 = 10.0;
458     final double itemRadius228 = 10.0;
459     final double itemElevation229 = 5.0;
460     final double itemWidth230 = 150.0;
461     final double itemHeight230 = 150.0;
462     final double itemMargin230 = 10.0;
463     final double itemRadius231 = 10.0;
464     final double itemElevation232 = 5.0;
465     final double itemWidth233 = 150.0;
466     final double itemHeight233 = 150.0;
467     final double itemMargin233 = 10.0;
468     final double itemRadius234 = 10.0;
469     final double itemElevation235 = 5.0;
470     final double itemWidth236 = 150.0;
471     final double itemHeight236 = 150.0;
472     final double itemMargin236 = 10.0;
473     final double itemRadius237 = 10.0;
474     final double itemElevation238 = 5.0;
475     final double itemWidth239 = 150.0;
476     final double itemHeight239 = 150.0;
477     final double itemMargin239 = 10.0;
478     final double itemRadius240 = 10.0;
479     final double itemElevation241 = 5.0;
480     final double itemWidth242 = 150.0;
481     final double itemHeight242 = 150.0;
482     final double itemMargin242 = 10.0;
483     final double itemRadius243 = 10.0;
484     final double itemElevation244 = 5.0;
485     final double itemWidth245 = 150.0;
486     final double itemHeight245 = 150.0;
487     final double itemMargin245 = 10.0;
488     final double itemRadius246 = 10.0;
489     final double itemElevation247 = 5.0;
490     final double itemWidth248 = 150.0;
491     final double itemHeight248 = 150.0;
492     final double itemMargin248 = 10.0;
493     final double itemRadius249 = 10.0;
494     final double itemElevation250 = 5.0;
495     final double itemWidth251 = 150.0;
496     final double itemHeight251 = 150.0;
497     final double itemMargin251 = 10.0;
498     final double itemRadius252 = 10.0;
499     final double itemElevation253 = 5.0;
500     final double itemWidth254 = 150.0;
501     final double itemHeight254 = 150.0;
502     final double itemMargin254 = 10.0;
503     final double itemRadius255 = 10.0;
504     final double itemElevation256 = 5.0;
505     final double itemWidth257 = 150.0;
506     final double itemHeight257 = 150.0;
507     final double itemMargin257 = 10.0;
508     final double itemRadius258 = 10.0;
509     final double itemElevation259 = 5.0;
510     final double itemWidth260 = 150.0;
511     final double itemHeight260 = 150.0;
512     final double itemMargin260 = 10.0;
513     final double itemRadius261 = 10.0;
514     final double itemElevation262 = 5.0;
515     final double itemWidth263 = 150.0;
516     final double itemHeight263 = 150.0;
517     final double itemMargin263 = 10.0;
518     final double itemRadius264 = 10.0;
519     final double itemElevation265 = 5.0;
520     final double itemWidth266 = 150.0;
521     final double itemHeight266 = 150.0;
522     final double itemMargin266 = 10.0;
523     final double itemRadius267 = 10.0;
524     final double itemElevation268 = 5.0;
525     final double itemWidth269 = 150.0;
526     final double itemHeight269 = 150.0;
527     final double itemMargin269 = 10.0;
528     final double itemRadius270 = 10.0;
529     final double itemElevation271 = 5.0;
530     final double itemWidth272 = 150.0;
531     final double itemHeight272 = 150.0;
532     final double itemMargin272 = 10.0;
533     final double itemRadius273 = 10.0;
534     final double itemElevation274 = 5.0;
535     final double itemWidth275 = 150.0;
536     final double itemHeight275 = 150.0;
537     final double itemMargin275 = 10.0;
538     final double itemRadius276 = 10.0;
539     final double itemElevation277 = 5.0;
540     final double itemWidth278 = 150.0;
541     final double itemHeight278 = 150.0;
542     final double itemMargin278 = 10.0;
543     final double itemRadius279 = 10.0;
544     final double itemElevation280 = 5.0;
545     final double itemWidth281 = 150.0;
546     final double itemHeight281 = 150.0;
547     final double itemMargin281 = 10.0;
548     final double itemRadius282 = 10.0;
549     final double itemElevation283 = 5.0;
550     final double itemWidth284 = 150.0;
551     final double itemHeight284 = 150.0;
552     final double itemMargin284 = 10.0;
553     final double itemRadius285 = 10.0;
554     final double itemElevation286 = 5.0;
555     final double itemWidth287 = 150.0;
556     final double itemHeight287 = 150.0;
557     final double itemMargin287 = 10.0;
558     final double itemRadius288 = 10.0;
559     final double itemElevation289 = 5.0;
560     final double itemWidth290 = 150.0;
561     final double itemHeight290 = 150.0;
562     final double itemMargin290 = 10.0;
563     final double itemRadius291 = 10.0;
564     final double itemElevation292 = 5.0;
565     final double itemWidth293 = 150.0;
566     final double itemHeight293 = 150.0;
567     final double itemMargin293 = 10.
```

Gambar 4. Implementasi Bloc dengan optimasi
(2)

Gambar 5. Implementasi Bloc dengan optimasi (3)

Gambar 6. Implementasi Bloc dengan optimasi (4)

Kode di atas adalah implementasi dari bloc yang sudah dioptimasi, kode ini berfungsi untuk mengatur logika pemanggilan data secara bertahap melalui mekanisme paging. Pada bagian ini, variabel currentPage dan perPage digunakan untuk menentukan halaman data dan jumlah data yang diambil per pemanggilan API. Event FetchUserEvent, LoadMoreUsersEvent, dan RefreshUserEvent masingmasing diatur oleh handler yang berbeda untuk menangani pengambilan data awal, penambahan data saat

pengguna menggulir ke bawah (infinite scroll), serta pembaruan data (refresh).

Saat event FetchUserEvent dijalankan, Bloc mengubah state menjadi UserLoadingState() untuk menandakan bahwa data sedang dimuat. Selanjutnya, jika data berhasil diambil, state berubah menjadi UserLoadedState, yang berisi kumpulan data pengguna serta informasi apakah halaman terakhir telah tercapai (hasReachedMax).

Penerapan infinite scroll dan paging dilakukan pada kode tampilan utama. Tampilan ini menggunakan BlocBuilder untuk memantau perubahan state. Komponen utama yang digunakan yaitu Listview.builder yang menampilkan daftar pengguna berdasarkan data dari state

```
 1 // dependencies
 2 import 'package:flutter/material.dart';
 3 import 'package:provider/provider.dart';
 4 import 'package:optimization_state/app_bar.dart';
 5 import 'package:optimization_state/bloc/optimization_bloc.dart';
 6 import 'package:optimization_state/presentation/widgets/user_card.dart';
 7
 8 class OptimisationState extends StatefulWidget {
 9   final Stopwatch stopwatch;
10   bool showEndReward;
11
12   OptimisationState({this.stopwatch, this.showEndReward});
13
14   @override
15   _OptimisationState createState() => _OptimisationState();
16 }
17
18 class _OptimisationState extends State<OptimisationState> {
19   final Stopwatch stopwatch = Stopwatch();
20   bool showEndReward = false;
21
22   @override
23   void initState() {
24     super.initState();
25     stopwatch.start();
26     context.read<OptimizationBloc>().add(OptimizationEvent());
27   }
28
29   @override
30   void build(BuildContext context) {
31     return BlocBuilder<OptimizationBloc, OptimizationState>(
32       builder: (context, state) {
33         if (state.isCompleted) {
34           return const Scaffold(
35             body: Center(
36               child: CircularProgressIndicator(),
37             ),
38           );
39         } else {
40           return Scaffold(
41             body: Center(
42               child: Row(
43                 mainAxisAlignment: MainAxisAlignment.spaceEvenly,
44                 children: [
45                   Text('Time elapsed: ${stopwatch.elapsedMilliseconds}'),
46                   Text('Time limit: ${stopwatch.duration.inMilliseconds}'),
47                 ],
48               ),
49             ),
50           );
51         }
52       },
53     );
54   }
55 }
```

Gambar 7. Kode tampilan dengan optimasi

Gambar 8. Kode tampilan dengan optimasi (2)

Gambar 9. Kode tampilan dengan optimasi (3)

Pada saat pengguna mencapai bagian bawah daftar kondisi if (index == state.data.length) akan terpenuhi, sehingga otomatis akan memanggil LoadMoreUsersEvent() untuk memuat data tambahan dari halaman berikutnya. Proses ini dilakukan tanpa interaksi manual pengguna, sehingga tampilan terasa lebih dinamis dan efisien. Selain itu RefreshIndicator digunakan agar pengguna dapat memuat ulang daftar dengan melakukan swipe down.

Dengan penerapan teknik ini, aplikasi tidak lagi memuat seluruh data sekaligus, melainkan hanya menampilkan sebagian data terlebih dahulu, kemudian memuat halaman berikutnya ketika pengguna menggulir ke bawah. Pendekatan ini berhasil mengurangi penggunaan memori serta mempercepat waktu muat awal aplikasi.

4.3. Analisis Data

Penelitian ini bertujuan untuk menganalisis dan membandingkan performa aplikasi mobile versi full load data dengan versi yang menerapkan optimasi. Pengujian dilakukan dengan 15 kali percobaan sebelum optimasi setelah optimasi. Data yang digunakan yaitu waktu memuat (ms), penggunaan memori (MB), dan responsivitas (FPS).

4.3.1. Waktu Memuat

Tabel 1. Hasil Uji Waktu Memuat

Uji	pretest	posttest
1	2468	1290
2	2360	1472
3	2421	1223
4	2800	1231
5	2859	1243
6	2442	1227
7	2497	1244
8	2847	1250
9	2818	1377
10	2437	1247
11	2858	1255
12	2695	1241
13	2456	1231
14	2790	1235
15	2427	1245
Rata-rata	2611,67	1267,40

Tabel 1 merupakan data hasil uji waktu memuat aplikasi dalam mili detik (ms) sebanyak 15 kali sebelum dan sesudah optimasi menggunakan paging dan infinite scroll. Dapat diketahui rata-rata waktu memuat sebelum optimasi yaitu 2611,67 mili detik sedangkan setelah optimasi yaitu 1267,40 mili detik.

Tabel 2. Deskripsi Data Waktu Memuat

Uji	N	Standar Deviasi	Nilai Terendah	Nilai Tertinggi
Pretest	15	197,33	2360	2859
posttest	15	68,09	1224	1472

Dari Tabel 2 didapat standar deviasi untuk pretest atau sebelum optimasi Adalah 197,33 untuk nilai terendahnya adalah 2360 mili detik dan tertinggi nya Adalah 2859 mili detik. Sedangkan, setelah optimasi (posttest) standar deviasi 68,09, dengan nilai terendah 1224 mili detik dan tertingginya 1472 mili detik

a. Uji Normalitas

Tabel 3. Uji Normalitas Waktu Memuat

Uji	Shapiro-Wilk	
	statistic	p-value
pretest	0,812562	0,005353
posttest	0,614538	0,000035

Pada pengujian ini data di uji menggunakan metode Shapiro-Wilk dengan bantuan program python. Jika nilai p-value lebih besar dari 0,05 maka data terdistribusi dengan normal.

Dari Tabel 3 diketahui bahwa hasil uji normalitas untuk data pretest adalah $p = 0,005353$ dan untuk posttest adalah $p = 0,000035$. Karena kedua nilai tersebut kurang dari 0,05 maka dapat disimpulkan bahwa data tidak terdistribusi dengan normal baik pretest maupun posttest.

Karena hasil dari uji normalitas baik pretest maupun posttest tidak berdistribusi normal, maka pengujian hipotesis menggunakan uji non-parametrik, yaitu Wilcoxon Signed-Rank Test.

b. Uji Homogenitas

Tabel 4. Uji Homogenitas Waktu Memuat

Uji	Lavene Test	
	statistic	p-value

pretest	13,158302	0,001130
---------	-----------	----------

Dalam penelitian ini, uji homogenitas dilakukan dengan metode Levene Test menggunakan bantuan program Python. Jika nilai p-value menunjukkan angka di atas 0,05, maka data dinyatakan homogen.

Berdasarkan Tabel 4 diperoleh nilai p-value sebesar 0,001130 (lebih kecil dari 0,05), sehingga dapat disimpulkan bahwa data pretest dan posttest tidak homogen.

c. Uji Hipotesis

Tabel 5. Uji Hipotesis Waktu Memuat

Uji	Wilcoxon Signed rank test	
	statistic	p-value
pretest	0,000000	0,000061

Uji hipotesis dilakukan untuk mengetahui apakah terdapat perbedaan signifikan pada performa waktu memuat aplikasi sebelum dan sesudah optimasi melalui penerapan teknik paging dan infinite scroll.

Berdasarkan Tabel 4 nilai p-value adalah 0,000061 yang artinya lebih kecil dari nilai signifikansi yaitu 0,05, maka H_0 ditolak. Sehingga dapat disimpulkan bahwa terdapat perbedaan yang signifikan antara waktu memuat sebelum dan sesudah optimasi. Hal ini menunjukan bahwa penerapan teknik paging dan infinite scroll meningkatkan efisiensi waktu memuat data dalam aplikasi.

4.3.2. Penggunaan Memori

Tabel 6. Hasil Uji Penggunaan Memori (MB)

Uji	pretest	posttest
1	20,2	9,8
2	19,8	8,8
3	19,1	9,1
4	21,7	9,7
5	17,8	9,7
6	19,4	9,8
7	19,5	9,0
8	17,5	8,5
9	19,1	9,7
10	17,5	8,3
11	16,6	9,7
12	20,5	8,6
13	21,2	8,4
14	20,0	7,7

15	17,7	9,8
Rata-rata	19,17	9,11

Tabel 6 merupakan hasil pengujian penggunaan memori (dalam satuan MB) sebelum dan sesudah optimasi sebanyak 15 kali. Secara umum penggunaan memori setelah optimasi menunjukkan penurunan yang konsisten dibandingkan sebelum dioptimasi.

Tabel 7. Deskripsi Data Penggunaan Memori

Uji	N	Standar Deviasi	Nilai Terendah	Nilai Tertinggi
Pretest	15	1,48	16,6	21,7
posttest	15	0,69	7,7	9,8

Dari Tabel 7 menunjukan standar deviasi sebelum optimasi 1,48 dengan nilai terendah 16,6 dan nilai tertinggi 21,7. Sedangkan, setelah dioptimasi nilai deviasi 0,69 dengan nilai terendah 7,7 dan nilai tertinggi 9,8.

a. Uji Normalitas

Tabel 8. Uji Normalitas Penggunaan Memori

Uji	Shapiro-Wilk	
	statistic	p-value
pretest	0,961519	0,718770
posttest	0,866130	0,029616

Uji normalitas ini dilakukan untuk memastikan apakah data penggunaan memori terdistribusi dengan normal. Pengujian ini dilakukan pada data pretest dan posttest menggunakan metode Shapiro-Wilk dengan bantuan program python.

Dari Tabel 8 nilai p-value pretest yaitu 0,718770 yang berarti lebih besar dari taraf signifikansi (0,05), maka data pretest terdistribusi dengan normal. Namun, pada p-value dari posttest yaitu 0,029616 yang lebih kecil dari 0,05 sehingga data posttest tidak terdistribusi dengan normal. Karena salah satu data tidak terdistribusi dengan normal maka uji hipotesis menggunakan uji non-parametrik, yaitu Wilcoxon Signed-Rank Test.

b. Uji Homogenitas

Tabel 9. Uji Homogenitas Penggunaan Memori

Uji	Lavene Test	
	statistic	p-value

pretest	5,706915	0,023878
---------	----------	----------

Uji homogenitas bertujuan untuk mengetahui apakah varian data pretest dan posttest pada penggunaan memori memiliki kesamaan.

Pada Tabel 9 hasil uji homogenitas menggunakan lavene test dengan program python nilai p-value adalah 0,023878 yang berarti lebih kecil dari 0,05 sehingga data pretest dan posttest penggunaan memori tidak homogen.

c. Uji hipotesis

Tabel 10. Uji Hipotesis Penggunaan Memori

Uji	Wilcoxon Signed rank test	
	statistic	p-value
pretest	0,000000	0,000061

Berdasarkan Tabel 10 hasil uji Wilcoxon signed rank test menghasilkan nilai p-value 0,000061 yang artinya lebih kecil dari nilai signifikansi yaitu 0,05 sehingga H_0 ditolak. Maka, disimpulkan bahwa terdapat perbedaan yang signifikan antara penggunaan memori sebelum dan sesudah optimasi menggunakan paging dan infinite scroll, sehingga berdampak positif pada efisiensi performa aplikasi.

4.3.3. Responsivitas

Tabel 11. Hasil Uji Responsivitas (FPS)

Uji	pretest	posttest
1	60	60
2	60	59
3	59	60
4	59	60
5	60	59
6	59	60
7	59	60
8	59	60
9	60	59
10	60	60
11	60	60
12	60	59
13	60	60
14	60	60
15	60	60
Rata-rata	59,67	59,73

Responsivitas menunjukkan seberapa lancar aplikasi berjalan secara visual. Berdasarkan hasil uji responsivitas menunjukkan nilai FPS relative tinggi dan stabil di angka 59 hingga 60 baik pretest maupun posttest.

Tabel 12. Deskripsi Data Responsivitas

Uji	N	Standar Deviasi	Nilai Terendah	Nilai Tertinggi
Pretest	15	0,49	59	60
posttest	15	0,46	59	60

Pada Tabel 12 menunjukkan standar deviasi sebelum optimasi 0,49 dengan nilai terendah 59 dan nilai tertinggi 60. Sedangkan, setelah dioptimasi nilai deviasi 0,46 dengan nilai terendah 59 dan nilai tertinggi 60. Hal ini menunjukkan bahwa teknik optimasi yang digunakan tidak menurunkan kualitas visual aplikasi, dan performa FPS tetap stabil dalam kedua kondisi.

a. Uji Normalitas

Tabel 13. Uji Normalitas Responsivitas

Uji	Shapiro-Wilk	
	statistic	p-value
pretest	0,603427	0,000027
posttest	0,560898	0,000011

Uji Normalitas ini menggunakan metode Shapiro-Wilk, dengan proses analisis dilakukan menggunakan program python.

Berdasarkan Tabel 13 diperoleh nilai p-value untuk pretest adalah 0,000027 dan untuk posttest adalah 0,000011 yang artinya keduanya kurang dari 0,05 sehingga data responsivitas tidak terdistribusi dengan normal. Maka untuk uji hipotesis akan dilakukan menggunakan metode Wilcoxon Signed-Rank

b. Uji Homogenitas

Tabel 14 Uji Homogenitas Responivitas

Uji	Lavene Test	
	statistic	p-value
pretest	0,148936	0,792471

Uji Homogenitas ini 38 menggunakan metode Levene dan dianalisis dengan bantuan program Python.

Hasil pengujian pada Tabel 14 menunjukkan p-value sebesar 0,792471, yang berarti lebih besar dari 0,05. Maka dapat

disimpulkan bahwa data responsivitas pretest dan posttest adalah homogen.

c. Uji Hipotesis

Tabel 15. Uji Hipotesis Responsivitas

Uji	Wilcoxon Signed rank test	
	statistic	p-value
pretest	20,000000	0,738883

Karena sebelumnya data pretest dan posttest tidak terdistribusi normal, maka pengujian hipotesis untuk mengetahui perbedaan responsivitas sebelum dan sesudah optimasi dilakukan menggunakan metode Wilcoxon Signed-Rank.

Berdasarkan Tabel 15 didapat nilai p-value adalah 0,738883 yang artinya lebih besar dari 0,05. Maka H_0 diterima. Hal ini menunjukkan bahwa tidak terdapat perbedaan yang signifikan pada responsivitas sebelum dan sesudah optimasi.

5. KESIMPULAN

- Hasil uji Wilcoxon signed rank test pada waktu memuat menunjukkan p-value 0,000061 (lebih kecil dari 0,05), yang artinya terdapat perbedaan yang signifikan dalam waktu memuat antara sebelum dan sesudah optimasi Perbaikan ini terjadi karena setelah optimasi data hanya dimuat sebagian, sehingga mengurangi beban dan waktu memuat aplikasi.
- Hasil uji wilcoxon signed-rank test pada penggunaan memori menghasilkan pvalue 0,000061 yang artinya lebih kecil dari 0,05. Hal ini menunjukkan perbedaan signifikan dalam penggunaan memori. Efisiensi ini dicapai karena versi optimasi hanya menyimpan sebagian data dalam memori secara bertahap tidak langsung seluruh data.
- Responsivitas aplikasi tetap terjaga dengan baik pada sebelum maupun sesudah optimasi. Uji Wilcoxon Signed-Rank menunjukkan tidak ada perbedaan signifikan dalam responsivitas dengan nilai p-value 0,738883 (lebih besar dari 0,05). Hal ini disebabkan karena ListView.builder() pada Flutter hanya merender widget yang berada dalam

viewport, sehingga performa rendering tetap optimal terlepas dari jumlah total data yang ada.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Allah SWT., kedua orang tua, dosen dan teman-teman yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] I. Nur Ramadhan and G. Saraswati Wilujeng, “Penerapan Database Redis Sebagai Optimalisasi Pemrosesan Kueri Data Pengguna Aplikasi SIRENSMA Berbasis Laravel,” *Technomedia Journal*, vol. 8, no. 3, pp. 64–77, Nov. 2023, doi: 10.33050/tmj.v8i3.2152.
- [2] T. Das, S. Prof, and M. Di Penta, “Investigating Performance Issues in Mobile Apps,” Gran Sasso Institute of Science, 2020. Accessed: Sep. 25, 2024. [Online]. Available: <https://iris.gssi.it/handle/20.500.12571/9686>
- [3] J. Pibernik, J. Dolić, L. Mandić, and V. Kovač, “Mobile-Application Loading-Animation Design and Implementation Optimization,” *Applied Sciences*, vol. 13, no. 2, p. 865, Jan. 2023, doi: 10.3390/app13020865.
- [4] R. L. B. Baptista, “FRAMEDROP-MOBILE CLIENT,” 2023.
- [5] M. A. Hawari, “Kanji Visual Sebagai Multimedia Pembelajaran Kanji N4 Pada Smartphone Berbasis Android,” 2020. Accessed: Sep. 27, 2024. [Online]. Available: <http://elibrary.unikom.ac.id/id/eprint/3150>
- [6] D. Saputra and R. Kania, “Implementasi Design Thinking untuk User Experience Pada Penggunaan Aplikasi Digital,” *Prosiding Industrial Research Workshop* ..., 2022, [Online]. Available: <https://jurnal.polban.ac.id/ojs-3.1.2/proceeding/article/view/4210>
- [7] I. Husain, P. Purwantoro, and C. Carudin, “ANALISIS PERFORMA STATE MANAGEMENT PROVIDER DAN GETX PADA APLIKASI FLUTTER,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 2, pp. 1417–1422, Sep. 2023, doi: 10.36040/jati.v7i2.6867.
- [8] Asri Putri Dwi Gita Andini, Dian Wahyuningsih, and Mahmud Yunus, “Analisis Dan Peningkatan Performa Aplikasi Berbasis Website Menggunakan Stress Tools Gtmetrix,” *TEMATIK*, vol. 9, no. 2, pp. 191–201, Dec. 2022, doi: 10.38204/tematik.v9i2.1071.

- [9] J. O. Rixen *et al.*, “The Loop and Reasons to Break It: Investigating Infinite Scrolling Behaviour in Social Media Applications and Reasons to Stop,” *Proc ACM Hum Comput Interact*, vol. 7, no. MHCI, pp. 1–22, Sep. 2023, doi: 10.1145/3604275.
- [10] F. D. Anastasia and I. V. Paputungan, “IMPLEMENTASI BLOC PATTERN PADA PENGEMBANGAN FRONTEND FITUR TOP UP LINKAJA APLIKASI M-BANKING AGEN46 DENGAN TEKNOLOGI FLUTTER (STUDI KASUS : PT. BANK NEGARA INDONESIA TBK),” *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, vol. 2, no. 1, Sep. 2022, doi: 10.20885/snati.v2i1.14.
- [11] L. S. L. Lestari, F. M. Mahardika, and D. I. Junaedi, “RANCANG BANGUN PROTOTIPE APLIKASI MOBILE SKORING DAN DASHBOARD HASIL UJIAN KERTAS BERBASIS FLUTTER,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 3S1, Oct. 2025, doi: 10.23960/jitet.v13i3S1.8058.
- [12] R. Lo *et al.*, “Penggunaan Bahasa Pemrograman Python dalam Menganalisis Hubungan Kualitas Kopi dengan Lokasi Pertanian Kopi,” *Jurnal Publikasi Teknik Informatika*, vol. 2, no. 2, pp. 100–109, 2023.
- [13] R. A. Siroj, W. Afgani, F. Fatimah, D. Septaria, and G. Z. Salsabila, “Metode penelitian kuantitatif pendekatan ilmiah untuk analisis data,” *Jurnal Review Pendidikan dan Pengajaran (JRPP)*, vol. 7, no. 3, pp. 11279–11289, 2024.
- [14] A. Adil *et al.*, “Metode Penelitian Kuantitatif dan Kualitatif: Teori dan Praktik,” *Jakarta: Get Press indonesia*, 2023.
- [15] H. Sihotang, “Metode penelitian kuantitatif,” 2023, *Uki Press*.
- [16] Dr. Rr. N. Fauziyah, *Analisis Data Menggunakan Uji Non Parametrik di Bidang Kesehatan Masyarakat dan Klinis*. Politeknik Kesehatan Kemenkes Bandung, 2020. [Online]. Available: <https://repo.poltekkesbandung.ac.id/7614/1/BUKU%20ANALISIS%20DENGAN%20UJI%20NON%20PARAMETRIK.pdf>
- [17] R. W. E. Yani, E. Permatasari, and Y. Armiyanti, “BIOSTATISTIKA,” 2023, *UPT PENERBITAN UNIVERSITAS JEMBER*. [Online]. Available: https://repository.unej.ac.id/jspui/bitstream/123456789/112781/1/Buku%20Ajar%20Biostatistika_Prof.%20Risty_fix_removed.pdf