JUKIM
ADMI Jurnal Ilmiah Multidisiplin

JURNAL JUKIM Vol 3 No. 2 Maret 2024 | P-ISSN: 2829-0488 E-ISSN: 2829-0518, Hal. 103-112

ALERT SYSTEM DESIGN MENGGUNAKAN TEKNOLOGI FIREBASE CLOUD MESSAGING (FCM)

Alex Kristian[1], Nuryuliani[2], Arif Rachman Hakim[3]
[1, 2] Fakultas Teknologi Industri, Universitas Gunadarma
[3]Puslitbang BMKG

Article History
Received : 16-Febuari-2024
Revised : 17- Febuari-2024
Accepted : 30-Maret-2024
Published : 31-Maret-2024

Corresponding author*:
Alex Kristian

Contact:
alexkristian@student.gunadarma.ac.id

Cite This Article:
Kristian, A. ., Nuryuliani, N., & Hakim, A. R. . (2024). ALERT SYSTEM DESIGN MENGGUNAKAN TEKNOLOGI FIREBASE CLOUD MESSAGING (FCM). Jurnal Ilmiah Multidisiplin, 3(02), 103–112.

DOI:
https://doi.org/10.56127/jukim.v3i02.1601

Abstract: *This research discusses the applications design to provide real-time disaster warning notifications through the Alert System feature using Firebase Cloud Messaging (FCM) technology. In this application, an Alert will appear automatically when a disaster occurs, close the screen of other applications that used by the user, so the users can immediately evacuate. Other main features are navigation to the nearest evacuation point based on the user's location, emergency action guides according to the type of disaster, and access to the latest disaster information via the Get Info page. The architecture of this application is based on Model-View-View Model (MVVM) which separates business logic from the user interface, and uses Firebase Cloud Messaging (FCM) and Geo Firestore technology to manage location data and notifications efficiently. Testing was carried out on various endpoints to ensure accurate and fast responses, as well as on the user interface to ensure ease of use and proper functionality.*
Keywords: *Firebase cloud Messaging, Model view ViewModel, Natural Disaster, Alert system*

Abstrak: Pada penelitian ini membahas tentang perancangan Aplikasi untuk memberikan notifikasi peringatan bencana secara real-time melalui fitur Alert System menggunakan teknologi Firebase Cloud Messaging (FCM). Dalam aplikasi ini pada saat terjadi bencana, maka akan muncul Alert yang secara otomatis menutup layar aplikasi lain yang sedang digunakan oleh pengguna, sehingga pengguna dapat segera melakukan evakuasi. Fitur utama lainnya adalah navigasi ke titik evakuasi terdekat berdasarkan lokasi pengguna, panduan tindakan darurat yang disesuaikan dengan jenis bencana, dan akses ke informasi bencana terkini melalui halaman Get Info. Arsitektur aplikasi ini didasarkan pada Model-View-View Model (MVVM) yang memisahkan logika bisnis dari antarmuka pengguna, serta menggunakan teknologi Firebase Cloud Messaging (FCM) dan GeoFirestore untuk mengelola data lokasi dan notifikasi secara efisien. Pengujian dilakukan terhadap berbagai endpoint untuk memastikan respons yang akurat dan cepat, serta terhadap antarmuka pengguna untuk menjamin kemudahan penggunaan dan fungsionalitas yang tepat.
Kata Kunci: Firebase cloud Messaging, Model view , Bencana Alam, Alert system

PENDAHULUAN

Ada beberapa bencana yang diakibatkan oleh peristiwa atau serangkaian peristiwa yang disebabkan oleh alam diantaranya gempa bumi, tsunami, gunung meletus, banjir, kekeringan, angin topan, dan tanah longsor[1]. Bencana alam yang diakibatkan oleh faktor alam sulit untuk diprediksi kapan peristiwa tersebut akan terjadi. Oleh sebab itu, pencegahan bencana alam dilakukan melalui himbauan peringatan dini kepada masyarakat yang dikirim melalui perangkat mobile. Agar peringatan tidak terlewatkan, maka perlu di rancang suatu Sistem himbauan yang dapat menutup layar perangkat jika pengguna sedang menggunakan perangkat mobile. Dengan adanya himbauan peringatan dini masyarakat dalam kondisi siap siaga sesuai dengan tingkatan ancaman bencana yang terjadi.

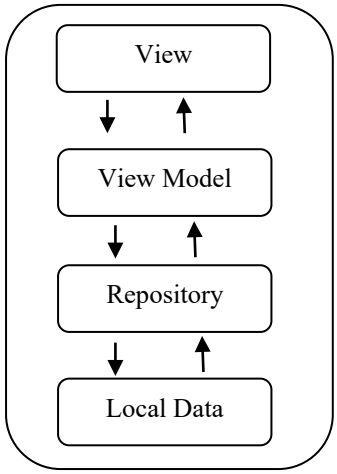
Aplikasi mengenai Bencana alam sudah dibuat oleh pemerintah bernama InaRisk. Ina risk adalah portal hasil kajian yang menggunakan acgis server sebagai data services yang menggambarkan cakupan wilayah ancaman bencana, populasi terdampak, potensi kerugian fisik, potensi kerugian ekonomi, dan potensi kerusakan lingkungan, dan terintegrasi dengan realisasi pelaksanaan kegiatan pengurangan risiko bencana sebagai tool monitoring penurunan indeks risiko bencana. InaRisk dibuat oleh BPNB dengan kontribusi dari UNDP, UNESCO, dan perkumpulan lainnya (BNPB)[2]. Namun aplikasi InaRISK belum memiliki sistem notifikasi himbauan kepada masyarakat, oleh karena itu penulis merancang sebuah sistem untuk memberikan himbauan kepada masyarakat.

Pada penulisan ini akan dikembangkan aplikasi mobile menggunakan Model-View-ViewModel (MVVM) dan teknologi Firebase Cloud Messaging (FCM). Aplikasi ini dirancang untuk memberikan notifikasi kepada pengguna ketika terjadi bencana di sekitar lokasi mereka, serta memberikan informasi terkait lokasi evakuasi terdekat. Dengan adanya aplikasi ini, diharapkan masyarakat dapat menerima informasi secara cepat dan tepat waktu, sehingga dapat mengambil tindakan yang diperlukan untuk menjaga keselamatan diri dan orang sekitarnya, serta membantu mengurangi risiko kerugian yang ditimbulkan oleh bencana alam melalui peningkatan kesadaran masyarakat dan akses informasi yang lebih cepat. MVVM juga mendukung bidirectional data binding antara komponen View dan ViewModel dan memiliki kinerja yang baik.[4, 5]

Pengembangan aplikasi mobile untuk mitigasi bencana merupakan langkah proaktif dalam pemanfaatan teknologi informasi untuk kepentingan kemanusiaan dan menjadi sarana yang efektif untuk memudahkan proses evakuasi.

METODE PENELITIAN

Pada penulisan ini menggunakan arsitektur MVVM (Model-View-ViewModel) karena pendekatan ini memudahkan pemisahan logika bisnis dari tampilan, sehingga meningkatkan modularitas dan keteraturan kode. MVVM mempermudah pengembangan, pemeliharaan, dan pengujian aplikasi, karena setiap komponen memiliki tanggung jawab yang jelas dan terpisah. Dengan MVVM, pengelolaan state dan data menjadi lebih terstruktur, memungkinkan pengembang untuk fokus pada masing-masing aspek aplikasi tanpa khawatir akan kompleksitas keseluruhan. Selain itu, MVVM mendukung reuse kode, sehingga mempercepat proses pengembangan aplikasi.



Gambar 3.1. Arsitektur MVVM

Dalam arsitektur MVVM terbagi atas beberapa layer, yaitu Model, View, dan ViewModel. Berikut adalah penjelasan ketiga layer tersebut

Membangun Repository

Repository berfungsi sebagai penghubung antara logika bisnis dan ViewModel dalam arsitektur MVVM. Di dalam repository, kumpulan fungsi-fungsi logika bisnis diorganisir dengan baik, yang biasanya melibatkan interaksi dengan ApiService untuk mengambil data dari API eksternal, serta Preferences untuk mengakses data yang tersimpan secara lokal.

Membangun View Model

Initial data binding dibuat menggunakan GetX pada saat proyek dijalankan. Binding ini berfungsi sebagai ViewModel dalam arsitektur MVVM, yang bertujuan untuk menginisialisasi dan mengelola state serta logika bisnis yang dibutuhkan oleh view secara efisien. Binding diimplementasikan dengan perintah Get.lazyPut, yang digunakan untuk mendaftarkan instance ViewModel hanya ketika diperlukan, sehingga mengoptimalkan penggunaan memori dan sumber daya aplikasi

Membangun Logika Bisnis

Penggunaan ApiService dan Preferences untuk memenuhi kebutuhan proyek ini. ApiService berperan dalam mengambil data dari API eksternal, memungkinkan aplikasi untuk berkomunikasi dengan *server* dan mendapatkan informasi yang diperlukan. Sementara itu, Preferences digunakan untuk menyimpan data tertentu secara lokal ke dalam *file*.

Membangun Firebase Cloud Messaging (FCM)
Firebase Cloud Messaging (FCM) digunakan dalam aplikasi untuk memungkinkan pengelolaan notifikasi dan eksekusi kode di latar belakang. Dengan integrasi FCM, aplikasi dapat tetap menerima notifikasi dan menjalankan kode tertentu meskipun dalam kondisi *terminated* (mati) atau sedang berjalan di latar belakang. Hal ini memastikan bahwa aplikasi tetap responsif terhadap pesan penting, seperti peringatan bencana atau informasi kritis lainnya, bahkan ketika tidak aktif secara langsung oleh pengguna.

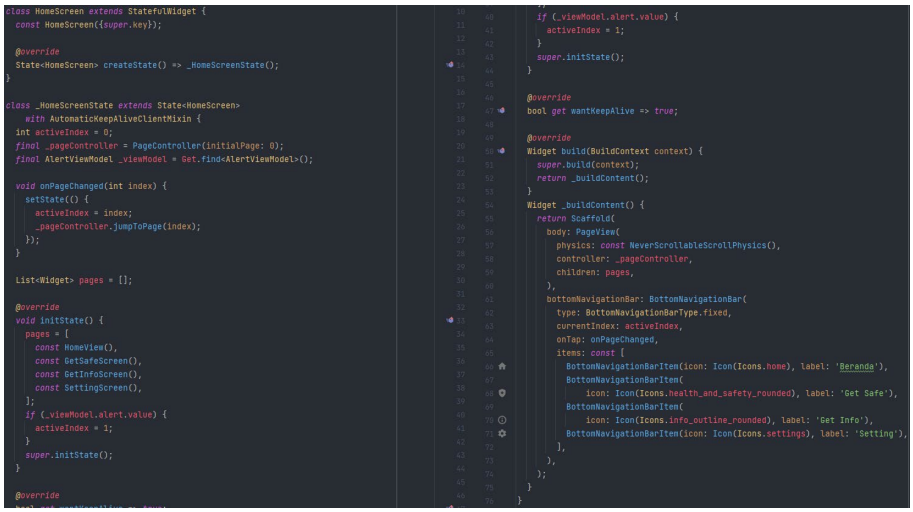
Membangun View
Tahap berikutnya adalah membangun beberapa tampilan (view) yang akan ditampilkan kepada pengguna dalam antarmuka aplikasi. Berikut adalah beberapa tampilan yang dikembangkan.

- Halaman Onboarding, dirancang untuk memberikan gambaran umum mengenai aplikasi kepada pengguna saat pertama kali menginstallnya, seperti dapat dilihat pada gambar 3.5 sbb:

```
import 'package:beamer/beamer.dart';
import 'package:colorful_safe_area/colorful_safe_area.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:get_safe/res/text/app_color.dart';
import 'package:get_safe/ui/onboarding/onboarding_view_model.dart';
import 'package:lottie/lottie.dart';
import 'package:onboarding/onboarding.dart';

final onboardImage = [
  "assets/images/onboard/onboard_1.png",
  "assets/images/onboard/onboard_2.png",
  "assets/images/onboard/onboard_3.png"
];

final onboardHeader = [
  "Alert System",
  "Informasi Bencana",
  "Menavigasi ke Mitigasi"
];

final onboardBody = [
  "Aplikasi get safe memiliki alert system yang dapat menghentikan aktifita
  "Aplikasi Get Safe memberikan panduan bencana alam ketika terjadi disekit
  "Ketika terjadi bencana, aplikasi membantu pengguna ke tempat daerah miti
];

class OnboardingScreen extends StatefulWidget {
  const OnboardingScreen({super.key});

  @override
  State<OnboardingScreen> createState() => _OnboardingScreenState();
}

class _OnboardingScreenState extends State<OnboardingScreen> {
  late int index;
  final OnboardingViewModel _vm = Get.find<OnboardingViewModel>();
```

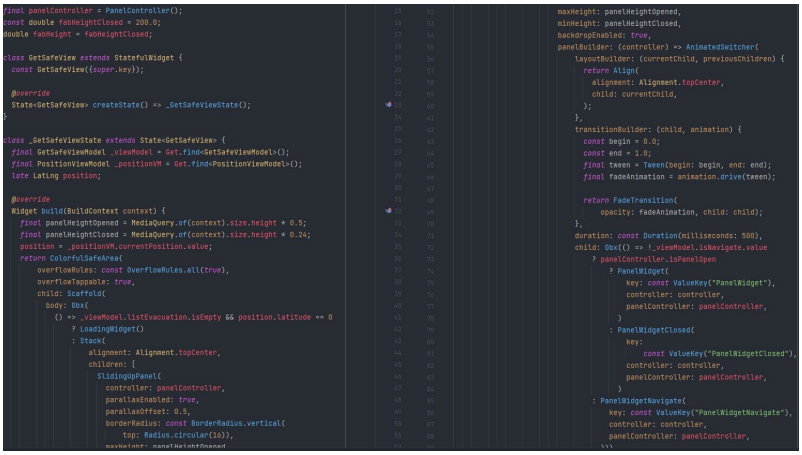
Gambar 3.5. rancangan Onboarding

- Halaman Home : Halaman *Home* menampilkan berbagai informasi penting, termasuk informasi gempa terkini yang diperbarui secara real-time, titik evakuasi terdekat dari lokasi pengguna, serta panduan *survival guide* untuk menghadapi bencana, , seperti dapat dilihat pada gambar 3.6 sbb:



Gambar 3.6. rancangan Halaman Home

- Halaman Get Safe, menampilkan peta interaktif yang menampilkan titik-titik evakuasi terdekat dari lokasi pengguna saat halaman dibuka, , seperti dapat dilihat pada gambar 3.7 sbb:.



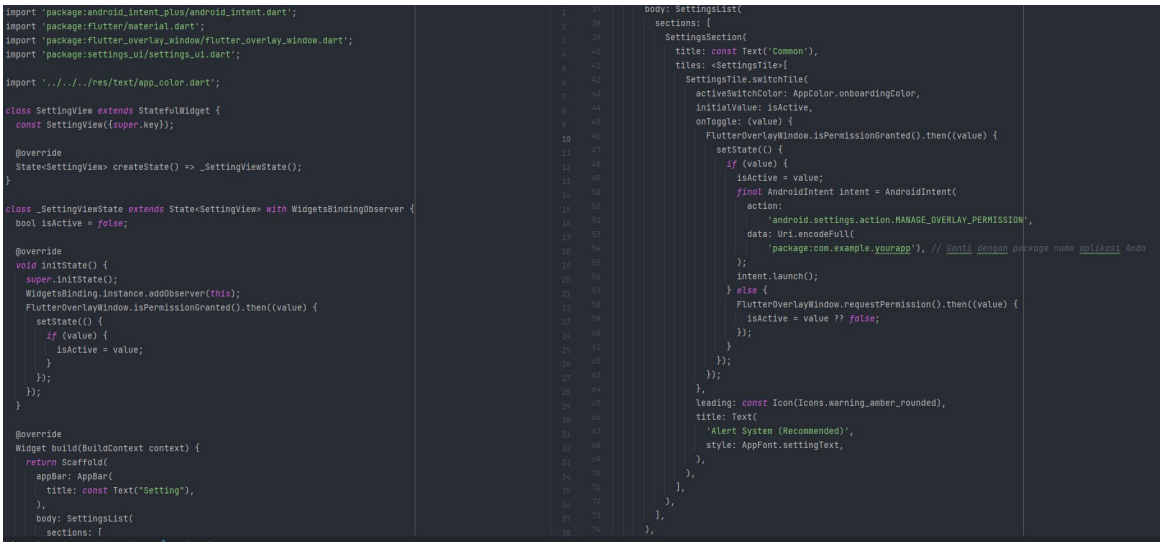
Gambar 3.7. rancangan Halaman Get Safe

- Halaman Get Info, menyajikan dengan informasi mengenai bencana terkini, , seperti dapat dilihat pada gambar 3.8 sbb:

```
class GetInfoView extends StatefulWidget {
  const GetInfoView({super.key});

  @override
  State<GetInfoView> createState() => _GetInfoViewState();
}

class _GetInfoViewState extends State<GetInfoView> {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: DefaultTabController(
        length: 3,
        child: Scaffold(
          appBar: AppBar(
            bottom: const TabBar(
              labelColor: AppColor.primaryColor,
              indicatorColor: AppColor.primaryColor,
              tabs: [
                Tab(text: "Gempa Bumi"),
                Tab(text: "Tsunami"),
                Tab(text: "Gunung Meletus"),
              ],
            ),  // TabBar
            centerTitle: true,
            title: const Text('Get Info'),
          ),  // AppBar
          body: const TabBarView(
            physics: NeverScrollableScrollPhysics(),
            children: [
              MapInfoView(
                pages: 0,
              ),  // MapInfoView
              MapInfoView(
```

Gambar 3.8. Halaman Get Info

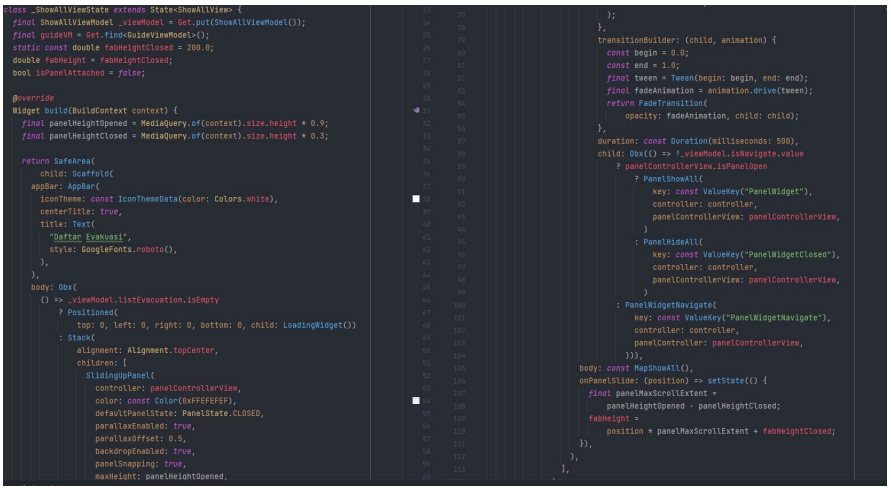
- Halaman Setting, pengguna dapat mengakses pengaturan dari aplikasi `Get Safe`. Halaman ini memungkinkan pengguna untuk mengaktifkan atau menonaktifkan sistem peringatan (alert system) sesuai dengan preferensi pengguna, , seperti dapat dilihat pada gambar 3.9 sbb:.



```
import 'package:android_intent_plus/android_intent.dart';
import 'package:flutter/material.dart';
import 'package:flutter_overlay_window/flutter_overlay_window.dart';
import 'package:settings_ui/settings_ui.dart';

import '../../../res/text/app_color.dart';

class SettingView extends StatefulWidget {
  const SettingView({super.key});

  @override
  State<SettingView> createState() => _SettingViewState();
}

class _SettingViewState extends State<SettingView> with WidgetsBindingObserver {
  bool isActive = false;

  @override
  void initState() {
    super.initState();
    WidgetsBinding.instance.addObserver(this);
    FlutterOverlayWindow.isPermissionGranted().then((value) {
      setState(() {
        if (value) {
          isActive = value;
        }
      });
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Setting"),
      ),
      body: SettingsList(
        sections: [
          SettingsSection(
            title: const Text('Common'),
            tiles: <SettingsTile>[
              SettingsTile.switchTile(
                activeSwitchColor: AppColor.onboardingColor,
                initialValue: isActive,
                onToggle: (value) {
                  FlutterOverlayWindow.isPermissionGranted().then((value) {
                    setState(() {
                      if (value) {
                        isActive = value;
                        final AndroidIntent intent = AndroidIntent(
                          action:
                              'android.settings.action.MANAGE_OVERLAY_PERMISSION',
                          data: Uri.encodeFull(
                              'package:com.example.yourapp'), // Ganti dengan package name aplikasi Anda
                        );
                        intent.launch();
                      } else {
                        FlutterOverlayWindow.requestPermission().then((value) {
                          isActive = value ?? false;
                        });
                      }
                    });
                  });
                },
                leading: const Icon(Icons.warning_amber_rounded),
                title: Text(
                  'Alert System (Recommended)',
                  style: AppFont.settingText,
                ),
              ),
            ],
          ),
        ],
      ),
    );
  }
```

Gambar 3.9. rancangan Halaman Setting

- Pada halaman *Show All Evacuation*, pengguna dapat melihat peta yang menampilkan semua titik evakuasi yang tersedia dalam aplikasi. Halaman ini dirancang untuk memberikan gambaran menyeluruh mengenai lokasi-lokasi evakuasi, memungkinkan pengguna untuk mengeksplorasi dan mengetahui semua opsi tempat evakuasi yang ada, , seperti dapat dilihat pada gambar 3.10 sbb:

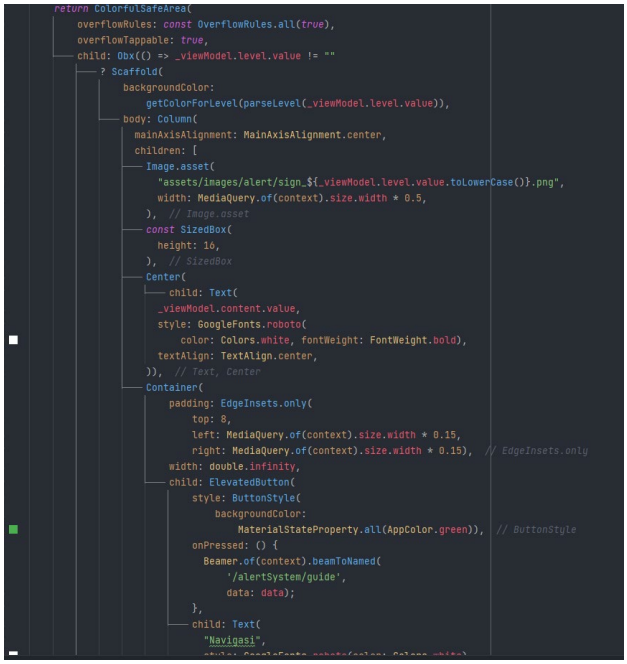


Gambar 3.10 rancangan Halaman Show All Evacuation

- Halaman Survival Guide, pengguna disajikan panduan mitigasi bencana yang mencakup beberapa skenario penting, seperti tindakan yang perlu diambil sebelum, saat, dan setelah terjadi bencana, , seperti dapat dilihat pada gambar 3.11 sbb:

Gambar 3.11. rancangan Halaman Survival Guide

- Halaman Alert System, pengguna akan diarahkan ke tampilan khusus yang muncul setiap kali notifikasi bencana diterima. Halaman ini dirancang untuk langsung menutup layar aktif pengguna dan menginterupsi aktivitas mereka, memastikan bahwa pengguna segera menyadari adanya situasi darurat, , seperti dapat dilihat pada gambar 3.12 sbb:



Gambar 3.12. rancangan Halaman Alert System

HASIL DAN PEMBAHASAN

Pengujian Black Box

Pengujian Black Box untuk memastikan bahwa sistem himbauan bencana alam berfungsi sesuai dengan spesifikasi yang telah ditentukan dan memenuhi kebutuhan pengguna. Pengujian black box pada sisi UI dilakukan dengan memeriksa fungsionalitas antarmuka tanpa melihat ke dalam kode sumber. Pengujian ini melibatkan verifikasi bahwa setiap elemen UI berfungsi sesuai dengan yang diharapkan, seperti tombol yang dapat diklik, navigasi yang berjalan lancar, input yang diterima dengan benar, dan respon aplikasi terhadap aksi pengguna. Penulis memeriksa apakah setiap fitur yang terlihat oleh pengguna bekerja sesuai spesifikasi, memastikan bahwa interaksi pengguna dengan aplikasi memberikan hasil yang benar dan konsisten, serta tidak ada bug atau kesalahan yang terlihat pada antarmuka.

Tabel 1 Pengujian Black Box UI

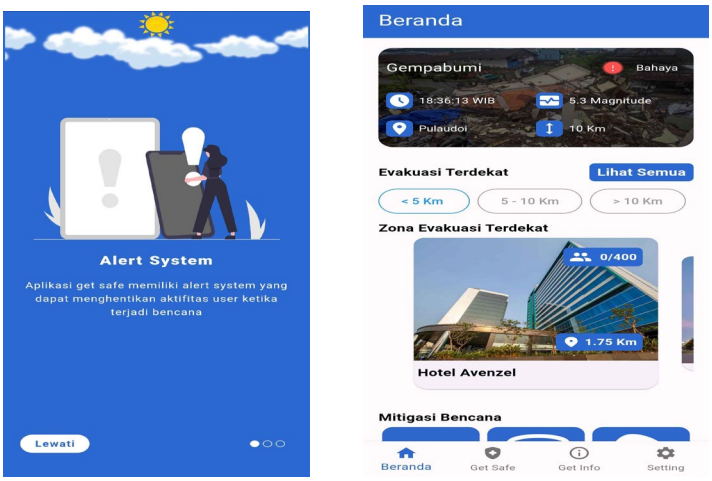
Data Masukan	Hasil yang diharapkan	Hasil Pengujian
Mengaktifkan System Alert pada Setting	Dapat menampilkan System Alert ketika terdapat notifikasi	Berhasil
Melihat mitigasi bencana	Dapat menampilkan mitigasi bencana	Berhasil
Melihat titik evakuasi	Dapat menampilkan titik evakuasi	Berhasil
Melihat informasi bencana terkini	Dapat menampilkan informasi bencana terkini	Berhasil
Melihat informasi 30 kejadian gempa terkini	Dapat menampilkan informasi 30 kejadian gempa terkini	Berhasil
Melihat informasi 30 kejadian gunung meletus terkini	Dapat menampilkan 30 kejadian gunung meletus terkini	Berhasil
Melihat informasi 30 kejadian tsunami terkini	Dapat menampilkan 30 kejadian tsunami terkini	Berhasil

Tabel 1 Pengujian Black Box Server

Data Masukan	Hasil yang diharapkan	Hasil Pengujian
Memasukkan data dan judul, body notification	Dapat melakukan notifikasi	Berhasil
Memasukkan data titik evakuasi	Dapat memasukkan data titik evakuasi ke server	Berhasil
Mengambil data titik evakuasi berdasarkan id	Dapat mengambil data titik evakuasi berdasarkan id	Berhasil
Mengambil semua data titik evakuasi	Dapat mengambil semua data titik evakuasi berdasarkan *area* nya.	Berhasil

Tampilan Aplikasi

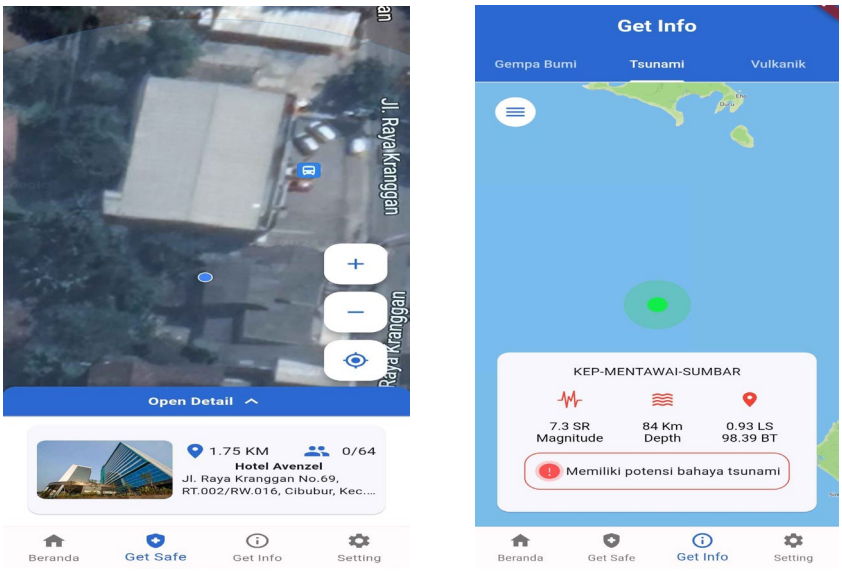
- Tampilan Halaman Onboarding, berisi gambaran umum mengenai aplikasi kepada pengguna saat pertama kali menginstallnya. Halaman Home : Halaman Home menampilkan berbagai informasi penting, termasuk informasi gempa terkini yang diperbarui secara real-time, titik evakuasi terdekat dari lokasi pengguna, serta panduan survival guide untuk menghadapi bencana dapat dilihat pada Gambar 4.1 sbb:



(a) (b)

Gambar 1. Tampilan Onboarding (a) dan Halaman Home (b)

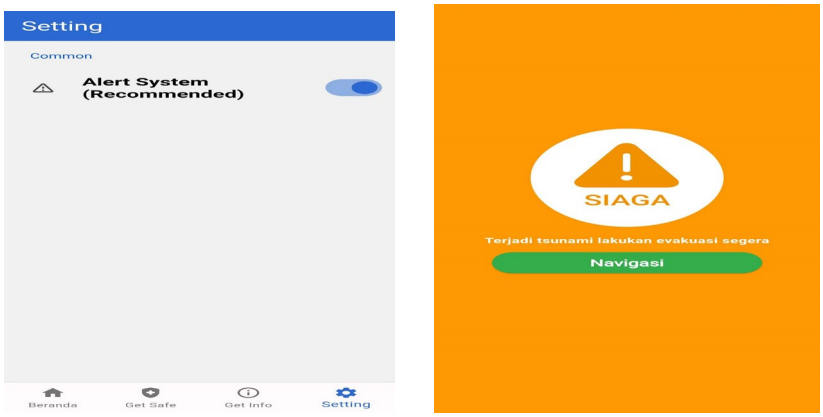
- Halaman Get Safe, menampilkan peta interaktif yang menampilkan titik-titik evakuasi terdekat dari lokasi pengguna saat halaman dibuka. Halaman Get Info, menyajikan dengan informasi mengenai bencana terkini, tampilan halaman dapat dilihat pada Gambar 4.2 sbb:



(a) (b)

Gambar 2. Tampilan Halaman Get Safe dan Gambar Halaman Get Info

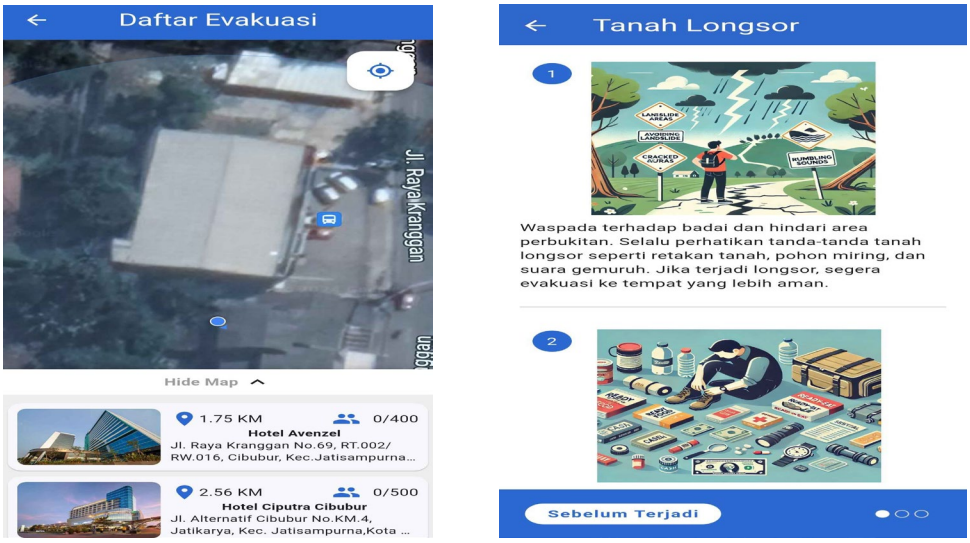
- Halaman Setting, pengguna dapat mengakses pengaturan dari aplikasi `Get Safe`. Halaman ini memungkinkan pengguna untuk mengaktifkan atau menonaktifkan sistem peringatan (alert system) sesuai dengan preferensi mereka. Halaman Alert System, pengguna akan diarahkan ke tampilan khusus yang muncul setiap kali notifikasi bencana diterima. Halaman ini dirancang untuk langsung menutup layar aktif pengguna dan menginterupsi aktivitas mereka, memastikan bahwa pengguna segera menyadari adanya situasi darurat, tampilan halaman dapat dilihat pada Gambar 4.3 sbb :



(a) (b)

Gambar 3. Tampilan Halaman Setting dan Halaman Alert System

- Pada halaman Show All Evacuation, pengguna dapat melihat peta yang menampilkan semua titik evakuasi yang tersedia dalam aplikasi. Halaman ini dirancang untuk memberikan gambaran menyeluruh mengenai lokasi-lokasi evakuasi, memungkinkan pengguna untuk mengeksplorasi dan mengetahui semua opsi tempat evakuasi yang ada. Halaman Survival Guide, pengguna disajikan panduan mitigasi bencana yang mencakup beberapa skenario penting, seperti tindakan yang perlu diambil sebelum, saat, dan setelah terjadi bencana, tampilan halaman dapat dilihat pada Gambar 4.4 sbb



(a) (b)

Gambar 4. Tampilan Halaman Show All Evacuation dan Halaman Survival Guide

KESIMPULAN

Aplikasi Alert System untuk bencana telah berhasil dikembangkan. Fitur ini memastikan bahwa pengguna langsung menyadari adanya bencana dan dapat mengambil tindakan cepat tanpa hambatan. Aplikasi dapat memberikan peringatan darurat yang efektif dengan cara menutup layar aplikasi pengguna saat berinteraksi dengan aplikasi lain, sehingga pengguna dapat segera melakukan evakuasi.
Fitur Alert System memberikan notifikasi real-time juga dilengkapi dengan navigasi ke titik evakuasi terdekat serta panduan tindakan darurat yang disesuaikan dengan kondisi saat bencana terjadi. Penerapan arsitektur MVVM dan teknologi seperti Firebase Cloud Messaging (FCM) mendukung kinerja dan responsivitas aplikasi, terutama dalam mengelola state dan notifikasi.

Secara keseluruhan, aplikasi ini tidak hanya berfungsi sebagai alat untuk mendukung evakuasi cepat dalam situasi darurat, tetapi juga sebagai platform edukatif yang membekali pengguna dengan pengetahuan dan kesiapan menghadapi bencana, sehingga dapat mengurangi risiko dan dampak negatif dari kejadian bencana.

Aplikasi ini dapat dikembangkan lebih lanjut seperti: Pengembangan Fitur Geofencing Untuk memperluas jangkauan notifikasi yang lebih akurat sesuai dengan lokasi geografis pengguna, terutama dalam situasi bencana yang cepat berubah dan juga Integrasi dengan Sumber Data Eksternal, seperti lembaga pemerintah atau organisasi internasional, untuk memberikan informasi yang lebih komprehensif dan up-to-date.

DAFTAR PUSTAKA

[1] Republik Indonesia. Undang-Undang Nomor 24 Tahun 2007 Tentang Penanggulangan Bencana.
[2] Badan Nasional Penanggulangan Bencana (BNPB). Tentang InaRISK. Tanggal Akses 17 Januari 2024, https://inarisk.bnpb.go.id/about
[3] Michael Stonis. 2022. Enterprise Application Patterns using .NET MAUI. Microsoft Developer Division. https://aka.ms/maui-ebook
[4] Akhmad Hilmy Zakaria, I Kadek Dwi Nuryana , MVVM Perangkat Lunak Android dan Analisis Arsitektur MVP dengan Studi Kasus Aplikasi iTourism, Journal of Informatics and Computer Science (JINACS): Volume 04 Nomor 04, 2023 , ISSN : 2686-2220 hal 351-359
[5] Mumuh Kustino Muharram, Zikri Ariachandra, Bambang Wisnuadhi, Ghifari Munawar, Analisis Pengaruh Arsitektur MVVM dan MVP pada Performa Database GreenDao, Prosiding The 12th Industrial Research Workshop and National Seminar Bandung, 4-5 Agustus 2021, hal 880-889