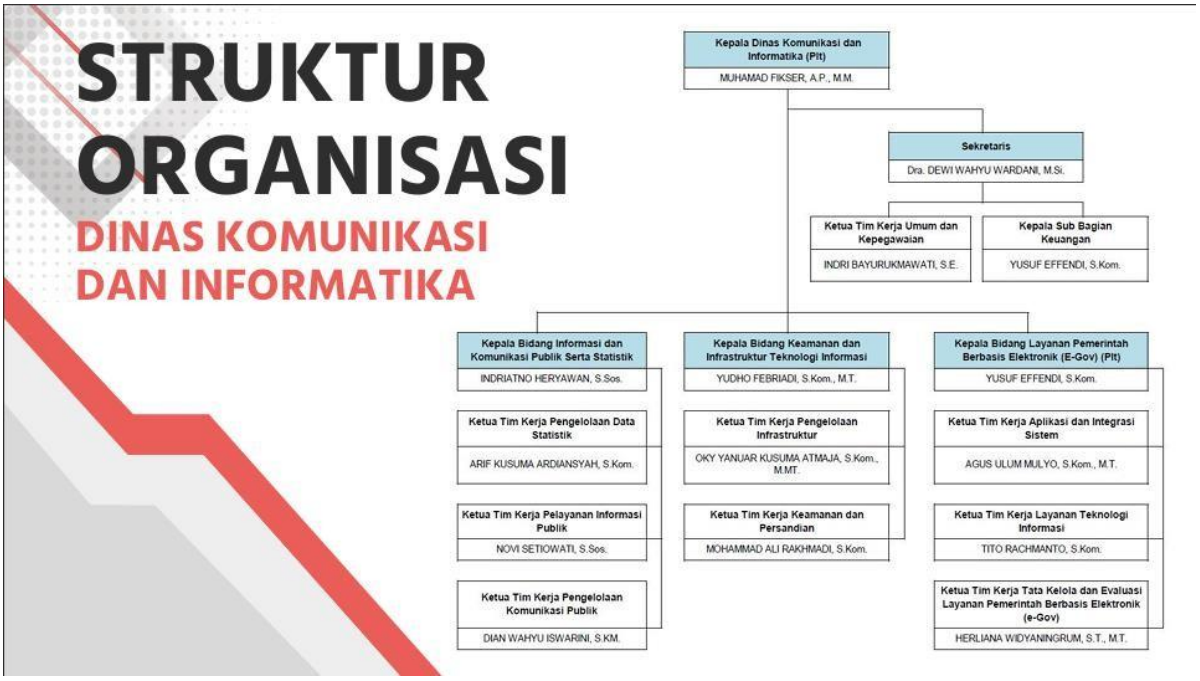
Bab II Gambaran Umum Tempat PKL

II.1 Struktur Organisasi



Gambar 2.1 Stuktur Organisasi Kominfo

Struktur Organisasi Dinas Komunikasi dan Informatika

Berdasarkan gambar, struktur organisasi Dinas Komunikasi dan Informatika terdiri dari beberapa bidang dan tim kerja dengan tugas dan tanggung jawab yang spesifik. Struktur organisasi ini dirancang untuk mendukung berbagai aspek komunikasi, informasi, teknologi, dan layanan publik.

Berikut adalah penjelasan struktur organisasi secara garis besar:

1. **Kepala Dinas**

   Kepala Dinas memiliki tanggung jawab untuk membuat kebijakan dan strategi di bidang komunikasi dan informatika yang sejalan dengan visi dan misi pemerintah daerah. Selain itu, ia memimpin tim dan mengoordinasikan seluruh pegawai di bawah naungannya, menjalin kerja sama dengan Organisasi Perangkat Daerah (OPD) lainnya, serta mengembangkan inovasi dan kreativitas dalam pengembangan sistem berbasis teknologi informasi. Ia juga bertugas mengatasi berbagai permasalahan yang muncul di bidang komunikasi dan informatika.

2. **Plt Bidang E-Goverment**

   Plt Bidang E-Government memiliki peran dalam merancang arsitektur, mengembangkan aplikasi, dan melakukan integrasi sistem. Ia bertanggung jawab atas implementasi dan pemeliharaan sistem agar selalu berjalan dengan baik, serta mengelola data dengan mengumpulkan, menganalisis, dan memastikan keamanannya. Dalam menjalankan tugasnya, ia juga berkoordinasi dan berkolaborasi dengan OPD lain, serta melakukan evaluasi dan pengembangan sistem secara berkelanjutan.

3. **Ketua Tim Kerja + Mentor**

   Sebagai Ketua Tim Kerja sekaligus Mentor, tugas utamanya meliputi memimpin, membuat rencana kerja, membagi tugas, dan memonitoring kinerja anggota tim. Ia juga mengidentifikasi masalah yang timbul, membangun hubungan antar anggota tim, dan melakukan koordinasi dengan departemen terkait guna memastikan kelancaran proyek. Selain itu, ia bertanggung jawab melaporkan hasil kerja kepada atasan atau pihak terkait.

4. **Tugas Mentor**

   Sebagai Mentor, tugasnya adalah memberikan bimbingan kepada peserta MSIB dalam menyelesaikan tugas mereka. Ia juga membagi tugas, membimbing peserta dalam pengembangan keterampilan teknis seperti penggunaan framework atau tools (contohnya Laravel, Flutter, dan One Signal Notification), serta membantu mereka membagi timeline proyek dan menetapkan target yang realistis. Mentor juga memberikan masukan untuk meningkatkan kualitas aplikasi, baik dalam aspek desain, tools, mekanisme penggunaan, maupun aspek fungsional.

5. **Tim Analis**

   Tim Analis memiliki tanggung jawab untuk mengidentifikasi kebutuhan secara mendalam terhadap sistem yang akan dibangun, menganalisis proses bisnis, dan membuat use case diagram. Mereka juga merancang alur kerja sistem, membuat prototipe, melakukan pengujian

sistem, serta menganalisis hasil pengujian tersebut. Evaluasi hasil kerja dilakukan secara berkala untuk memastikan sistem bekerja dengan baik.

6. **Database Administrator**

   Database Administrator bertugas merancang database dengan membangun entitas, atribut, dan relasi data untuk menciptakan struktur database yang efisien. Ia menentukan jenis database yang sesuai dengan kebutuhan sistem, mendesain skema database, membuat database di server, serta menguji apakah database tersebut sudah memenuhi kebutuhan. Selain itu, ia juga melakukan pemeliharaan database dengan backup rutin, recovery, dan optimasi, menjaga keamanan database dari akses tidak sah, memonitor aktivitas mencurigakan, serta mengembangkan fitur baru untuk mendukung sistem.

7. **Technical Writter**

   Technical Writer bertugas membuat dokumentasi seperti user manual yang berisi panduan lengkap penggunaan aplikasi dan panduan teknis untuk pengguna tingkat lanjut seperti admin. Ia juga membuat tutorial singkat untuk fitur baru, release notes untuk setiap pembaruan sistem, serta slide presentasi untuk menjelaskan fitur-fitur baru atau konsep teknis kepada audiens.

8. **Programmer**

   Programmer memiliki tanggung jawab membangun aplikasi sesuai spesifikasi yang dibutuhkan, mengintegrasikan sistem seperti database, payment gateway, dan autentikasi, serta mengembangkan fitur baru sesuai kebutuhan pengguna. Ia juga melakukan pemeliharaan aplikasi dengan memperbaiki bug, mengoptimasi kinerja aplikasi agar lebih cepat dan efisien, serta melakukan pembaruan keamanan untuk melindungi aplikasi dari serangan siber. Pengujian setiap bagian program secara individu juga menjadi bagian dari tugas programmer.Posisi Tim pada Struktur Organisasi

Selama pelaksanaan program MBKM, saya ditempatkan di bawah Bidang Layanan Pemerintah Berbasis Elektronik (E-Gov), lebih spesifik pada Tim Kerja

Aplikasi dan Integrasi Sistem yang dipimpin oleh Agus Ulum Mulyo, S.Kom., M.T.
Tim ini bertanggung jawab atas:
Pengembangan aplikasi berbasis Laravel untuk mendukung layanan internal dinas.
Pembuatan fitur-fitur integrasi pada sistem pemerintahan berbasis elektronik.
Penyelesaian masalah teknis pada aplikasi dan layanan berbasis teknologi lainnya.

II.2 Lingkup Pekerjaan

Selama mengikuti program MBKM, saya ditempatkan di **Bidang Layanan Pemerintah Berbasis Elektronik (E-Gov)** di Dinas Komunikasi dan Informatika. Bidang ini memiliki tanggung jawab utama dalam mengelola, mengembangkan, dan mengevaluasi berbagai layanan pemerintahan berbasis teknologi informasi. Lingkup pekerjaan divisi ini mencakup pengembangan aplikasi dan sistem yang bertujuan untuk mendukung efisiensi dan efektivitas layanan publik. Selain itu, bidang ini juga bertugas memastikan bahwa seluruh layanan yang disediakan sesuai dengan standar keamanan dan kualitas teknologi informasi terkini.

Secara spesifik, salah satu bagian penting dari pekerjaan divisi ini adalah **Tim Kerja Aplikasi dan Integrasi Sistem**, tempat saya berkontribusi selama pelaksanaan MBKM. Tim ini berfokus pada pengembangan dan implementasi sistem aplikasi berbasis web dan mobile yang memungkinkan integrasi data serta layanan antarunit kerja di dalam pemerintahan. Selain itu, tim ini juga bertugas mendukung penyelesaian masalah teknis yang muncul pada layanan berbasis teknologi, baik yang sudah berjalan maupun yang sedang dalam tahap pengembangan.

Lingkup pekerjaan saya selama berada di tim ini mencakup beberapa aspek penting yang selaras dengan tanggung jawab tim. Salah satu tugas utama adalah mengembangkan fitur-fitur aplikasi berbasis framework Laravel untuk kebutuhan internal dinas. Proyek yang saya kerjakan melibatkan pengelolaan database, pembuatan API untuk mendukung integrasi layanan antarplatform, dan pengembangan antarmuka pengguna (UI/UX) pada aplikasi berbasis web. Selain

itu, saya juga terlibat dalam pengembangan aplikasi Flutter untuk mendukung layanan mobile yang lebih responsif dan modern.

Seluruh pekerjaan ini didasarkan pada kebutuhan untuk meningkatkan kualitas layanan publik melalui teknologi yang lebih canggih dan efisien. Misalnya, proyek pengembangan aplikasi berbasis Laravel yang saya kerjakan bertujuan untuk mempermudah pengelolaan data administrasi pemerintahan. Proyek ini tidak hanya mencakup pembuatan fitur CRUD (Create, Read, Update, Delete) tetapi juga menambahkan fitur keamanan, autentikasi, dan integrasi API yang memungkinkan aplikasi bekerja secara sinergis dengan sistem lain di bawah naungan E-Gov.

Di sisi lain, proyek Flutter yang saya ikuti bertujuan untuk menghadirkan solusi mobile bagi masyarakat, seperti portal informasi berbasis aplikasi yang menampilkan berita atau informasi layanan publik. Aplikasi ini dikembangkan dengan desain responsif dan user-friendly untuk mempermudah akses masyarakat terhadap layanan digital yang disediakan oleh pemerintah.

Lingkup pekerjaan ini tidak hanya melibatkan aspek teknis tetapi juga melibatkan proses diskusi dan kolaborasi dengan tim lain untuk memastikan aplikasi yang dikembangkan dapat memenuhi kebutuhan unit kerja dan masyarakat secara luas. Dengan demikian, pengalaman saya dalam lingkup pekerjaan ini tidak hanya memberikan pemahaman teknis yang lebih dalam, tetapi juga wawasan tentang bagaimana teknologi dapat digunakan secara efektif untuk mendukung pelayanan publik dan pemerintahan berbasis elektronik.

II.3 Deskripsi Pekerjaan

Pengembangan Fitur CRUD dan Autentikasi di Laravel:

1. **Fitur CRUD untuk Data Pegawai, SPK, dan Pelatihan**:

Saya membuat tampilan form untuk memasukkan data pegawai, SPK, dan pelatihan, yang disesuaikan dengan struktur database. Form ini memungkinkan pengguna untuk mengisi data yang diperlukan, seperti informasi pegawai, nomor kontrak SPK, dan data pelatihan. Selain itu, saya mengatur struktur database dan

memastikan bahwa relasi antar tabel terjaga dengan baik. Kemudian, saya mengembangkan controller untuk menangani penyimpanan data ke dalam database, memastikan bahwa data yang dimasukkan dapat disimpan dengan benar dan sesuai dengan aturan yang telah ditentukan.

2. **Fitur Login**:

Untuk memastikan hanya pengguna yang terverifikasi yang dapat mengakses aplikasi, saya mengembangkan sistem **login** dengan menggunakan email dan password. Fitur ini dihubungkan dengan tabel **data_utama** di database, yang berisi informasi pengguna, termasuk kolom **email** dan **password**. Saya membuat rute, controller, dan view yang menangani proses autentikasi, termasuk pengecekan kredensial pengguna. Setelah berhasil login, pengguna akan diarahkan ke halaman utama aplikasi.

3. **Fitur Logout**:

Sebagai pelengkap fitur login, saya juga mengimplementasikan sistem **logout** yang memungkinkan pengguna keluar dari aplikasi dan mengakhiri sesi mereka. Fitur ini menggunakan session atau token untuk mengidentifikasi pengguna yang sedang aktif, dan setelah logout, pengguna akan diarahkan kembali ke halaman login

Pengembangan Aplikasi Mobile Wargaku 2:

Dalam proyek Wargaku 2, saya berhasil menyelesaikan hampir seluruh halaman utama aplikasi, dengan berbagai fitur yang telah dikembangkan, seperti:

- Halaman Home: Menyediakan menu-menu utama yang memungkinkan pengguna untuk menavigasi aplikasi dengan mudah.
- Halaman Berita: Halaman ini terhubung langsung dari halaman home dan menampilkan daftar berita dengan kategori yang berbeda.
- Profil Utama: Menampilkan informasi pengguna, seperti nama, foto profil, dan data pribadi lainnya.
- Navigasi ke Notifikasi dan Scan QR: Menyediakan fitur navigasi untuk mengakses notifikasi dan melakukan pemindaian QR dari halaman home.

Desain Responsif dengan MediaQuery:

Untuk memastikan tampilan aplikasi tetap rapi dan responsif pada berbagai ukuran layar, saya menggunakan MediaQuery di Flutter. Ini memungkinkan elemen-elemen di dalam aplikasi menyesuaikan diri dengan ukuran layar perangkat, seperti posisi dan ukuran widget.

Implementasi Bottom Navigation Bar:

Saya menambahkan Bottom Navigation Bar dengan opsi navigasi seperti Home, Profile, dan MyCard. Selama proses ini, saya juga melakukan debugging untuk memastikan tampilan navigasi yang lancar dan bebas dari masalah tampilan.

Integrasi API:

Sebagian besar fitur dalam aplikasi membutuhkan integrasi dengan API eksternal. Saya menghubungkan aplikasi dengan beberapa API, terutama untuk fitur registrasi pengguna, kategori berita, dan fitur forgot password. Data yang ditarik dari API digunakan untuk menampilkan informasi di aplikasi, dan proses ini berjalan lancar setelah saya mengonfigurasi koneksi dan menangani masalah yang muncul.

Pengelolaan State Menggunakan Provider:

Untuk mengelola data secara efisien di seluruh aplikasi, saya menggunakan Provider. Provider memungkinkan data seperti profil pengguna dan berita untuk terupdate secara global tanpa perlu memanggil ulang data di setiap widget. Dengan menggunakan Provider, saya dapat memastikan data yang ditampilkan di berbagai layar selalu terkini.

Pengembangan Fitur News di Flutter:

Pada fitur berita, saya membuat dua model utama: NewsCategory dan NewsData. Model-model ini digunakan untuk menyimpan data berita dan kategori yang terstruktur, serta memudahkan pemrosesan data JSON yang diterima dari API.

- Provider untuk Fetching Data:

  Untuk mengambil data berita dari API, saya menggunakan Provider dengan fetch yang bertugas untuk memanggil API dan mengelola state

data. Fungsi fetch dalam provider mengirimkan permintaan ke API dan kemudian memetakan data yang diterima ke dalam model yang sesuai.

- Service untuk Menghubungkan API:
  Di dalam aplikasi, saya membuat Service yang bertugas mengelola permintaan dan respons API. Service ini berfungsi sebagai penghubung antara aplikasi dan server, memastikan bahwa data ditarik dengan benar dan menghindari penanganan langsung API di dalam widget.
- Penggunaan Factory Constructor pada Model:
  Mengingat struktur data JSON yang cukup kompleks dan bersarang (nested), saya memanfaatkan factory constructor pada model untuk mempermudah parsing data. Dengan cara ini, saya dapat mengubah data JSON menjadi objek model Flutter yang mudah diolah.
- Loading Indicator dan Error Handling:
  Untuk meningkatkan pengalaman pengguna, saya menambahkan loading indicator saat aplikasi sedang menarik data berita dari API. Selain itu, saya juga menambahkan penanganan error handling menggunakan try-catch untuk menampilkan pesan kesalahan yang informatif jika terjadi masalah dengan koneksi atau jika data tidak berhasil dimuat.

WebView untuk Akses URL:

Saya juga mengembangkan fitur WebView dalam aplikasi. Fitur ini memungkinkan pengguna untuk mengakses URL dari halaman Home atau Berita dan melakukan berbagai navigasi, seperti:

- Next: Berpindah ke halaman berikutnya.
- Previous: Kembali ke halaman sebelumnya.
- Refresh: Memuat ulang halaman saat ini.
- Back: Kembali ke halaman sebelumnya.

Halaman Forgot Password:

Selain itu, saya juga mengembangkan halaman Forgot Password, yang memungkinkan pengguna mengirimkan permintaan untuk mereset password mereka. Halaman ini menerima parameter email dari pengguna dan mengirimkan data tersebut ke API eksternal yang menangani permintaan reset password. Setelah

pengguna mengirimkan email, sistem API akan mengirimkan instruksi lebih lanjut untuk melakukan reset password jika email terdaftar di sistem.

II.4 Jadwal Kerja

Tabel 3.1 Log Kegiatan

Minggu/Tgl	Kegiatan	Hasil
1/11-13 September	Membuat database untuk website pegawai, membuat relasi antar data di data model, serta menyiapkan controller untuk membuat fungsi menampilkan data, input data, dan edit data.	Database untuk website pegawai telah dibuat, relasi antar data di data model berhasil ditetapkan, dan controller yang berfungsi untuk menampilkan data, input data, dan mengedit data sudah disiapkan.
2/16-20 September	Pengenalan tugas dari mentor, mulai membuat navigasi, membuat fitur login yang dapat login dengan data yang akan diisi pada data utama, membuat tampilan untuk input data, dan membuat tampilan untuk menampilkan data.	Tugas dari mentor berhasil dikenalkan, navigasi mulai dibangun, fitur login yang menggunakan data dari tabel data utama sudah berfungsi, tampilan untuk input data telah dibuat, dan tampilan untuk menampilkan data sudah disiapkan.
3/23-27 September	Membuat tampilan untuk mengedit data, memulai pengembangan fitur input data SPK (Surat Perintah Kerja), input data Pelatihan, input data	Tampilan untuk mengedit data telah dibuat, pengembangan fitur input dan menampilkan data

	Peringatan, dan input data Perintah Tugas, memulai pengembangan fitur untuk menampilkan data SPK (Surat Perintah Kerja), input data Pelatihan, input data Peringatan, dan input data Perintah Tugas, membuat halaman profile yang akan menampilkan data yang telah diinput, serta membuat fitur untuk menampilkan halaman profile, dan membuat halaman home.	SPK, Pelatihan, Peringatan, dan Perintah Tugas telah dimulai, halaman profile untuk menampilkan data yang telah diinput sudah tersedia dengan fitur yang berfungsi, serta halaman home juga telah berhasil dibuat.
4/30-4 September-Oktober	Membuat fitur untuk logout, persiapan project baru Wargaku2 mobile dengan setup Android Studio dan mobile API, menyesuaikan template di Android Studio, membuat agenda kota (databasenya menggunakan JSON sementara), dan membuat fitur search pada berita.	Fitur logout berhasil dibuat, persiapan project Wargaku2 mobile selesai dengan setup Android Studio dan mobile API, template di Android Studio sudah disesuaikan, agenda kota berhasil dibuat dengan database menggunakan JSON sementara, dan fitur search pada berita sudah berfungsi dengan baik.
1/7-11 Oktober	Membuat model agenda kota, menghubungkan halaman agenda kota ke menu utama, membuat UI untuk halaman home dengan semua icon, membuat 2 halaman onboarding yang menjelaskan apa itu Wargaku dan apa saja kegunaannya, serta membuat fitur search pada halaman home.	Model agenda kota telah dibuat dan terhubung ke menu utama, UI untuk halaman home dengan semua icon berhasil diselesaikan, dua halaman onboarding yang menjelaskan tentang Wargaku dan

		kegunaannya sudah tersedia, dan fitur search pada halaman home sudah berfungsi dengan baik.
2/14-18 Oktober	Mengubah keseluruhan icon untuk halaman login, register, dan profile, menambahkan icon scan dan notifikasi serta menghubungkannya ke file Dart, membuat fitur sementara untuk scan QR code, membuat halaman bantuan, dan menambahkan fitur logout yang diarahkan kembali ke halaman login.	Icon untuk halaman login, register, dan profile berhasil diubah, icon scan dan notifikasi sudah ditambahkan dan terhubung ke file Dart, fitur sementara untuk scan QR code telah dibuat, halaman bantuan berhasil disiapkan, dan fitur logout yang mengarahkan kembali ke halaman login sudah berfungsi.
3/21-25 Oktober	Menambahkan widget NavBottom dengan tiga opsi navigasi: Home, Profile, Call 112, menambahkan kategori agenda kota seperti Trending, Popular, Politics, Art, dan Food, memperbaiki bug pada BottomNavigationBar yang tidak mengubah halaman secara dinamis dengan menambahkan setState(), membuat form registrasi dengan TextFormField untuk input email, NIK, nama, nomor telepon, dan password, mengintegrasikan API	Widget NavBottom dengan tiga opsi navigasi: Home, Profile, dan Call 112 sudah berhasil ditambahkan, kategori agenda kota seperti Trending, Popular, Politics, Art, dan Food telah ditambahkan, bug pada BottomNavigationBar yang sebelumnya tidak mengubah halaman

	untuk menampilkan daftar aplikasi menggunakan http package untuk permintaan GET, menghubungkan form registrasi ke endpoint POST request, dan menambahkan fungsi handleRegis() untuk memetakan data input form ke body request API, serta menampilkan respon sukses atau error menggunakan QuickAlert.	secara dinamis telah diperbaiki dengan menambahkan setState(), form registrasi dengan input email, NIK, nama, nomor telepon, dan password sudah disiapkan, API berhasil diintegrasikan untuk menampilkan daftar aplikasi melalui permintaan GET, form registrasi sudah terhubung ke endpoint POST request, dan fungsi handleRegis() berfungsi untuk memetakan data input form ke body request API dengan respon sukses atau error yang ditampilkan menggunakan QuickAlert.
4/28-1 Oktober-November	Menambahkan fungsi handleRegis() untuk memetakan data input form ke body request API dan menampilkan respon sukses atau error menggunakan QuickAlert, memastikan data wilayah seperti provinsi, kecamatan, dan kelurahan dikirim dengan benar ke API berdasarkan input pengguna, membetulkan fungsi kategori berita	Fungsi handleRegis() berhasil mengeksekusi pemetaan data input form ke body request API dan menampilkan respon sukses atau error dengan QuickAlert, data wilayah seperti provinsi, kecamatan, dan kelurahan

	baru seperti Trending, Popular, Politics, Art, dan Food pada model NewsCategory, melakukan testing fungsi registrasi dengan berbagai skenario seperti input tidak lengkap atau email sudah digunakan, serta menambahkan teks kebijakan dan privasi pada halaman registrasi yang mengharuskan pengguna menekan centang sebelum registrasi.	berhasil dikirim dengan benar ke API sesuai input pengguna, kategori berita baru seperti Trending, Popular, Politics, Art, dan Food pada model NewsCategory sudah diperbaiki, fungsi registrasi telah diuji dengan berbagai skenario, termasuk input tidak lengkap dan email yang sudah digunakan, dan teks kebijakan serta privasi telah ditambahkan pada halaman registrasi, dengan centang yang wajib ditekan sebelum pengguna dapat melanjutkan registrasi.
1/4-8 November	Menguji API endpoint secara menyeluruh, termasuk validasi parameter dan respons yang diterima, meninjau aplikasi untuk bug kecil seperti layout yang tidak responsif di perangkat tertentu, menambahkan fitur reset password menggunakan email, menghubungkan forgot password dengan API, dan mengetes apakah forgot password bekerja dengan baik.	API endpoint berhasil diuji secara menyeluruh dengan validasi parameter dan respons yang diterima sesuai harapan, bug kecil pada aplikasi, termasuk layout yang tidak responsif di perangkat tertentu, telah diperbaiki, fitur reset password menggunakan email berhasil

		ditambahkan dan terhubung dengan API, serta fitur forgot password sudah diuji dan berfungsi dengan baik.
2/11-15 November	Memperbaiki QuickAlert pada forgot password menggunakan respon yang diberikan API, memperbaiki splash screen agar bekerja dengan memeriksa status bearer token (jika sudah expired akan diarahkan ke halaman login, jika belum expired akan langsung terhubung ke halaman home), memperbaiki provider agar lebih teratur, menghubungkan UI pada halaman home dengan API yang berisi JSON, dan melakukan testing integrasi fitur utama (login, registrasi, berita) untuk memastikan semua bekerja tanpa kendala.	QuickAlert pada fitur forgot password berhasil diperbaiki agar menggunakan respon yang diberikan oleh API, splash screen sekarang bekerja dengan memeriksa status bearer token dan mengarahkan pengguna ke halaman login jika token sudah expired, atau ke halaman home jika token masih valid, provider telah diperbaiki agar lebih teratur dan efisien, UI pada halaman home berhasil dihubungkan dengan API yang mengandung JSON, dan testing integrasi fitur utama (login, registrasi, berita) menunjukkan bahwa semua fitur berjalan dengan baik tanpa kendala.

3/18-22 November	Meninjau aplikasi untuk bug kecil seperti layout yang tidak responsif di perangkat tertentu, melakukan uji aplikasi menggunakan Firebase, menambahkan fitur pencarian agenda kota berdasarkan kata kunci pada model dan provider, menyesuaikan kata-kata pada onboarding, dan memperbaiki padding pada bagian registrasi.	Bug kecil pada aplikasi, termasuk layout yang tidak responsif di perangkat tertentu, telah berhasil diperbaiki, uji aplikasi dengan Firebase dilakukan dengan baik, fitur pencarian agenda kota berdasarkan kata kunci pada model dan provider sudah ditambahkan dan berfungsi, kata-kata pada onboarding telah disesuaikan, serta padding pada bagian registrasi telah diperbaiki untuk tampilan yang lebih rapi.
4/25-29 November	Mengecek apakah API calendar agenda kota bekerja, membuat widget untuk menampilkan daftar berita menggunakan data dari NewsProvider, menambahkan navigasi kategori di atas daftar berita menggunakan fetchCategories, mengintegrasikan fetchNews agar berita dapat difilter berdasarkan kategori, dan memperbaiki bug pada cache key fetchNews agar mendukung parameter kategori.	API calendar agenda kota berhasil diperiksa dan berfungsi dengan baik, widget untuk menampilkan daftar berita menggunakan data dari NewsProvider telah dibuat dan berfungsi, navigasi kategori di atas daftar berita berhasil ditambahkan menggunakan fetchCategories,

		fetchNews berhasil diintegrasikan untuk memungkinkan berita difilter berdasarkan kategori, dan bug pada cache key fetchNews telah diperbaiki sehingga mendukung parameter kategori dengan benar.
1/2-6 November	Membuat tampilan horizontal untuk daftar kategori dengan ListView.builder, menambahkan log debug untuk memastikan ID kategori yang dipilih terdeteksi dengan benar, mengubah logika cache pada fetchNews agar lebih efisien dan menghindari pengambilan data berulang, memperbaiki tampilan jika daftar kategori kosong dengan menampilkan pesan "No categories available", menangani error jika data berita gagal dimuat dengan menampilkan pesan error ke pengguna, dan menghilangkan penggunaan variabel uniqueCategories agar lebih sederhana.	Tampilan horizontal untuk daftar kategori berhasil dibuat menggunakan ListView.builder, log debug telah ditambahkan untuk memastikan ID kategori yang dipilih terdeteksi dengan benar, logika cache pada fetchNews telah diperbaiki agar lebih efisien dan mencegah pengambilan data berulang, tampilan daftar kategori kosong sudah diperbaiki dengan menampilkan pesan "No categories available", error yang terjadi saat data berita gagal dimuat sudah ditangani dengan menampilkan pesan error

		yang sesuai ke pengguna, dan penggunaan variabel uniqueCategories telah dihilangkan untuk menyederhanakan kode.
2/9-13 Desember	Menambahkan validasi data pada metode fetchCategories, membuat pengaturan ulang kategori saat pengguna kembali ke halaman berita, memperbaiki tampilan error saat data kategori gagal dimuat, menambahkan pengurutan berita berdasarkan tanggal mulai (startDate), dan mengintegrasikan fitur pencarian berita berdasarkan judul dengan parameter search.	Validasi data pada metode fetchCategories berhasil ditambahkan untuk memastikan data kategori valid, pengaturan ulang kategori saat pengguna kembali ke halaman berita sudah diterapkan dengan benar, tampilan error saat data kategori gagal dimuat telah diperbaiki untuk menampilkan pesan yang lebih informatif, pengurutan berita berdasarkan tanggal mulai (startDate) telah berhasil diterapkan, dan fitur pencarian berita berdasarkan judul dengan parameter search berhasil diintegrasikan dan berfungsi dengan baik.
3/16-20	Menambahkan fitur Maps di berita, menguji aplikasi pada berbagai perangkat dan ukuran layar untuk memastikan tampilan responsif,	Fitur Maps berhasil ditambahkan di halaman berita, aplikasi telah diuji pada berbagai perangkat

	melakukan pengujian aplikasi secara menyeluruh untuk memastikan bahwa halaman berita berfungsi dengan baik, melakukan perbaikan pada desain halaman berita agar lebih user-friendly, dan menyesuaikan desain dengan menggunakan MediaQuery untuk memastikan tampilan responsif pada berbagai ukuran layar.	dan ukuran layar untuk memastikan tampilan responsif, pengujian aplikasi secara menyeluruh menunjukkan bahwa halaman berita berfungsi dengan baik, desain halaman berita telah diperbaiki agar lebih user-friendly, dan desain aplikasi disesuaikan menggunakan MediaQuery untuk memastikan tampilan responsif pada berbagai ukuran layar.
4/23-27 Desember	Menambahkan previous, next, dan refresh pada webview agenda kota, melakukan pengujian aplikasi dan memperbaiki bug tampilan dan data, serta finalisasi fitur berita untuk memastikan semua elemen berfungsi dengan baik.	Fitur previous, next, dan refresh berhasil ditambahkan pada webview agenda kota, aplikasi telah diuji dan bug tampilan serta data berhasil diperbaiki, dan fitur berita telah difinalisasi dengan memastikan semua elemen berfungsi dengan baik tanpa kendala.
5/30-31 Desember	Melakukan dokumentasi dan offboarding MSIB	Dokumentasi telah selesai disusun dengan baik, mencakup proses dan

		hasil yang dicapai selama periode MSIB, dan offboarding MSIB telah dilakukan dengan lancar, menyelesaikan semua prosedur administrasi dan memastikan transisi yang baik.