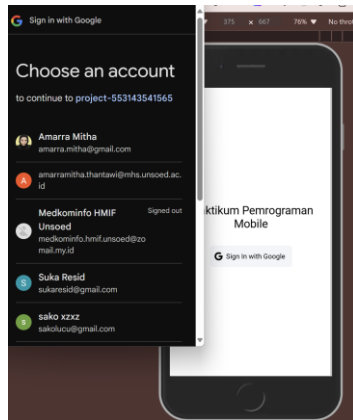


Nama : Amarramitha Poodja Thantawi
NIM : H1D022064
Shift Baru : C
Shift KRS : A

Responsi 2 Pemograman Mobile

1. Proses Login

- Saat pengguna mengklik tombol "Sign In with Google", fungsi `loginWithGoogle` dipanggil.
- Aplikasi menggunakan `isPlatform('capacitor')` untuk mendeteksi apakah aplikasi berjalan di platform mobile (iOS/Android) atau web.



- Aplikasi membuat instance `GoogleAuthProvider` untuk otentikasi Google di Firebase.
- Dengan `signInWithPopup`, Firebase memunculkan popup Google login, di mana pengguna dapat login menggunakan akun Google mereka.
Jika berhasil, Firebase mengembalikan objek `result` yang berisi data pengguna yang berhasil login.
- Setelah proses login berhasil (baik pada platform web atau mobile), `result.user` berisi informasi akun Google pengguna yang sedang login, seperti `displayName` (nama pengguna), `email`, `photoURL` (foto profil), dan lainnya.
- Data pengguna ini disimpan dalam variabel `user`, yang merupakan ref dengan tipe `User` dari Firebase, sehingga dapat diakses di seluruh aplikasi melalui `store auth`.

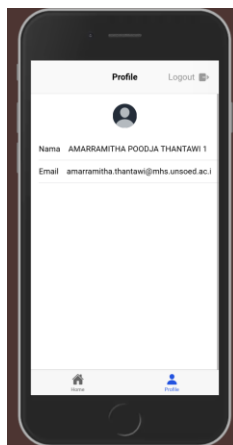
2. Navigasi ke halaman utama

- Setelah login berhasil, aplikasi akan melakukan navigasi ke halaman utama (`/home`) menggunakan `router.push("/home");`.



3. Mengakses nama pengguna dan profil

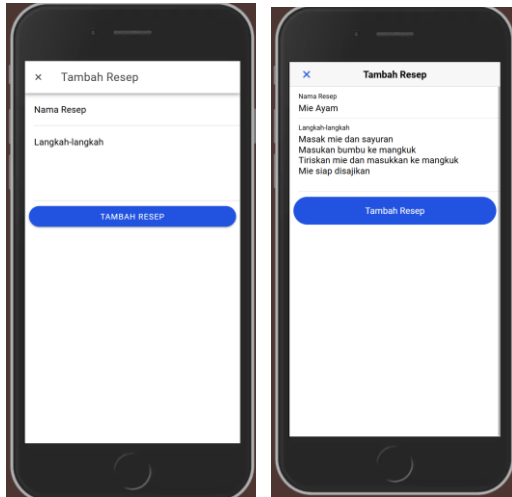
- Setelah pengguna berhasil login, data pengguna dari akun Google disimpan dalam objek user.
- Objek ini berisi informasi akun, seperti displayName (nama pengguna) dan photoURL (URL foto profil), yang dapat diakses melalui user.value.displayName dan user.value.photoURL.
- Aplikasi dapat menggunakan informasi ini untuk menampilkan nama pengguna dan foto profil di berbagai halaman atau komponen.



4. Proses Create

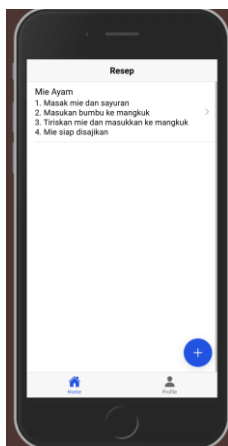
- Ketika tombol Tambah pada Floating Action Button (FAB) ditekan, modal input (InputModal) akan terbuka. Modal ini menerima data resep yang baru, dengan form input untuk nama resep dan langkah-langkah pembuatan resep.
- Pengguna mengisi form untuk nama resep dan langkah-langkahnya. Ketika tombol submit diklik, data tersebut akan dikirimkan melalui event **@submit** ke fungsi **handleSubmit**.

- Jika **editingId** adalah null (karena ini adalah resep baru), fungsi **addResep** dipanggil untuk menambahkan resep ke Firestore dengan informasi dari form input.
- Setelah data berhasil disimpan, modal ditutup, dan daftar resep diperbarui.



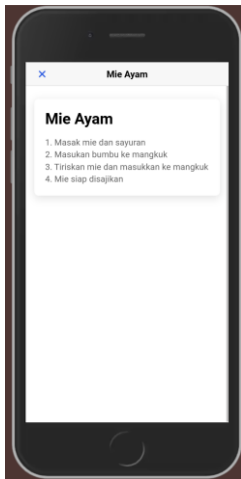
5. Proses Read

- Daftar resep ditampilkan menggunakan komponen `ion-list`. Setiap resep akan ditampilkan dengan nama dan langkah-langkah yang diformat (menggunakan **formattedStepsForList** untuk menampilkan langkah-langkah dengan pemisah baris).
- Ketika pengguna menekan item resep, fungsi **goToDetail** dipanggil untuk membuka modal detail. Modal ini menunjukkan detail resep yang lebih lengkap, termasuk nama resep dan langkah-langkah yang diformat dengan nomor urut.
- Modal ini dibuka dengan menggunakan properti **isModalOpen** yang diubah menjadi true. Pengguna dapat melihat resep secara lengkap di dalam modal tersebut.



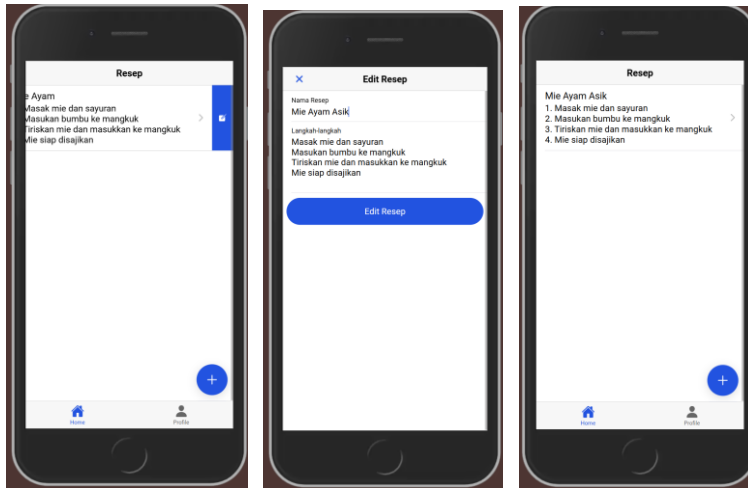
6. Halaman Detail Resep

- Modal detail ini digunakan untuk menampilkan informasi lengkap tentang resep yang dipilih, termasuk nama resep dan langkah-langkahnya yang diformat dengan nomor urut. Modal ini akan muncul di layar setelah **goToDetail** dipanggil.
- **currentResep**: Variabel ini menyimpan data resep yang dipilih (seperti nama dan langkah-langkah). Data ini digunakan untuk di-render dalam modal detail.
- **isModalOpen**: Variabel ini mengontrol apakah modal sedang ditampilkan atau tidak. Modal akan muncul jika nilainya true, dan hilang ketika nilainya false.
- Pengguna dapat menutup modal dengan menekan tombol "Tutup" yang ada di bagian header modal. Tombol ini akan memicu **isModalOpen** = false, yang akan menutup modal.



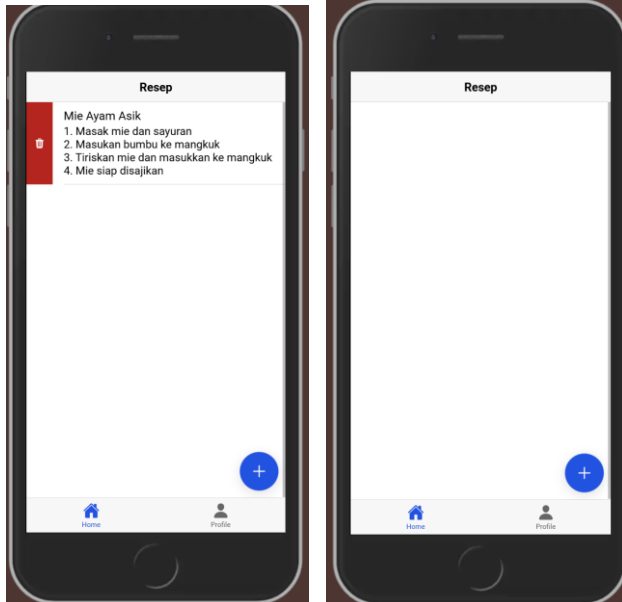
7. Proses Update

- Jika pengguna ingin mengedit resep, mereka menekan tombol edit yang ada pada opsi samping dari item resep di dalam ion-item-options. Fungsi **handleEdit** dipanggil.
- Fungsi **handleEdit** akan memuat data resep yang akan diedit, menyetel **editingId** dengan ID resep yang dipilih, dan menampilkan modal input dengan data resep yang sudah ada.
- Modal input akan menampilkan data yang sudah ada untuk resep yang sedang diedit, memungkinkan pengguna untuk mengubahnya.
- Setelah pengguna mengubah data dan menekan tombol submit, data baru akan disimpan dengan memanggil fungsi **updateResep** untuk memperbarui resep yang ada di Firestore. Daftar resep akan diperbarui.



8. Proses Delete

- Pengguna dapat menghapus resep dengan menekan tombol hapus pada opsi samping item resep di dalam ion-item-options.
- Fungsi `handleDelete` akan dipanggil, yang menghapus resep dari Firestore dengan menggunakan `deleteResep`.
- Setelah resep dihapus, daftar resep diperbarui dengan memanggil `loadReseps` untuk memuat data resep terbaru.



Soal

1. Sebuah menu pada aplikasi memerlukan load data setiap kali menu tersebut diakses, pada fungsi apakah load data tersebut dilakukan untuk framework Angular dan Vue?

Jawab:

- Pada Angular, load data biasanya dilakukan pada **ngOnInit()**, yang merupakan lifecycle hook yang dipanggil sekali setelah komponen diinisialisasi. Ini adalah tempat yang ideal untuk mengambil data dari API atau sumber lain saat komponen pertama kali dimuat.
- Di Vue.js, load data biasanya dilakukan dalam lifecycle hook **mounted()**, yang dipanggil setelah komponen ditambahkan ke DOM. Fungsi ini sering digunakan untuk melakukan request API atau memuat data saat komponen siap digunakan.

2. Pada menu profil, komponen bawaan Ionic apa saja yang dapat diterapkan untuk membangun menu tersebut? (min. 3 dan sebutkan untuk apa fungsinya)

Jawab:

- **ion-card**: Komponen ini digunakan untuk menampilkan informasi yang terstruktur, seperti nama, foto profil, dan deskripsi pengguna dalam bentuk kartu yang menarik.
Fungsi: Menyajikan informasi profil pengguna dengan cara yang lebih visual dan terorganisir.
- **ion-avatar**: Komponen ini digunakan untuk menampilkan gambar profil pengguna dalam bentuk lingkaran.
Fungsi: Menampilkan foto profil pengguna yang biasanya digunakan dalam menu profil.
- **ion-item**: Komponen ini digunakan untuk menampilkan elemen menu atau pilihan dalam daftar. Dalam menu profil, ini dapat digunakan untuk menampilkan opsi seperti "Edit Profil", "Pengaturan", dll.
Fungsi: Menyediakan tampilan daftar interaktif di menu profil.

3. Pada firebase, apa perbedaan antara Cloud Firestore dan Realtime Database?

Jawab:

Cloud Firestore lebih terstruktur dengan dokumen dan koleksi, sedangkan Realtime Database menggunakan struktur pohon JSON. Firestore mendukung query lebih kompleks, konsistensi data lebih baik, dan lebih skalabel untuk aplikasi besar. Realtime Database cocok untuk aplikasi kecil hingga menengah, dengan harga berdasarkan

bandwidth dan penyimpanan, sementara Firestore dihitung berdasarkan operasi baca, tulis, dan hapus.