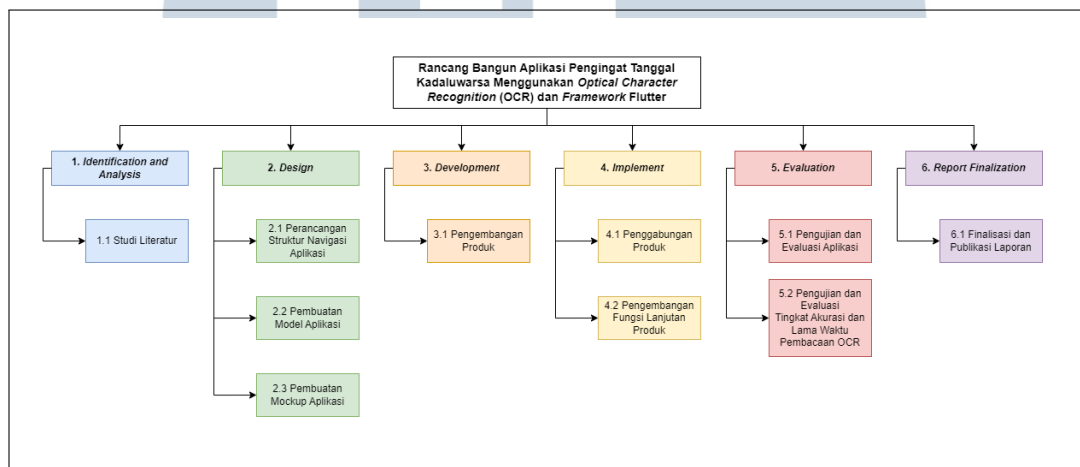


BAB 3 METODOLOGI PENELITIAN

Penelitian ini dirancang dan dibangun dengan menggunakan metode *Research and Development* dengan pendekatan metode ADDIE (*Analysis, Design, Development, Implement, Evaluation*). Diagram *Work Breakdown Structure* (WBS) pada Gambar 3.1 di bawah ini menunjukkan enam buah tahapan dan masing-masing aktivitas yang dilakukan selama penelitian ini dilangsungkan.

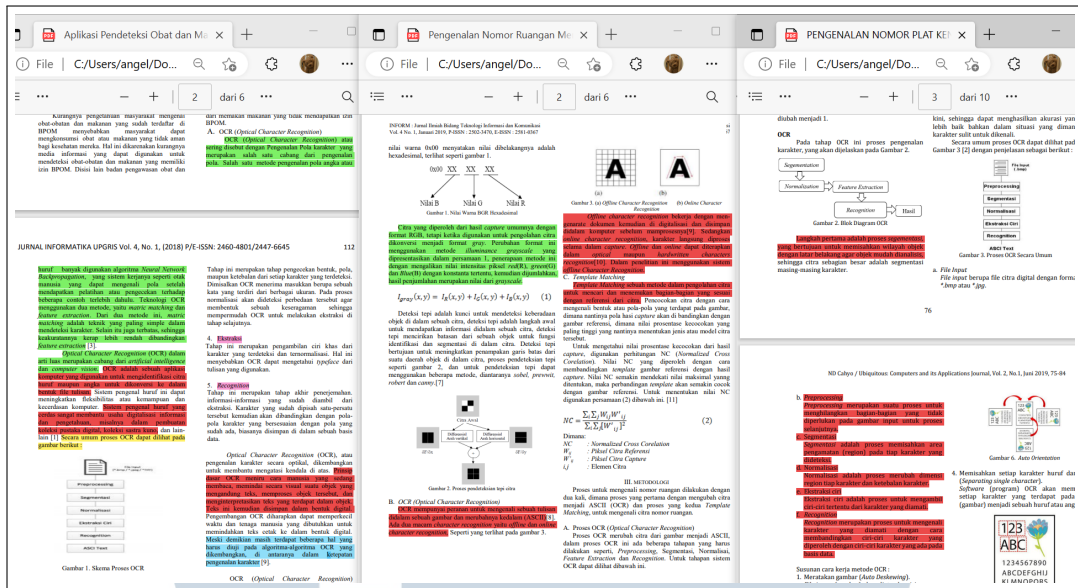


Gambar 3.1. Diagram *Work Breakdown Structure* (WBS)

Berdasarkan diagram WBS diatas, penelitian ini dilakukan berdasarkan tahapan dan aktivitas sebagai berikut.

3.1 *Identification and Analysis*

Tahapan ini dilakukan untuk memahami dan menganalisis topik penelitian dengan mencari berbagai sumber literatur untuk dijadikan sebagai bahan referensi. Sumber-sumber literatur yang telah ditemukan dapat dilihat pada bagian Daftar Pustaka dan informasi yang didapatkan serta dirangkum pada bagian Bab 2. Gambar 3.2 di bawah ini merupakan dokumentasi pengerjaan pada tahapan ini.



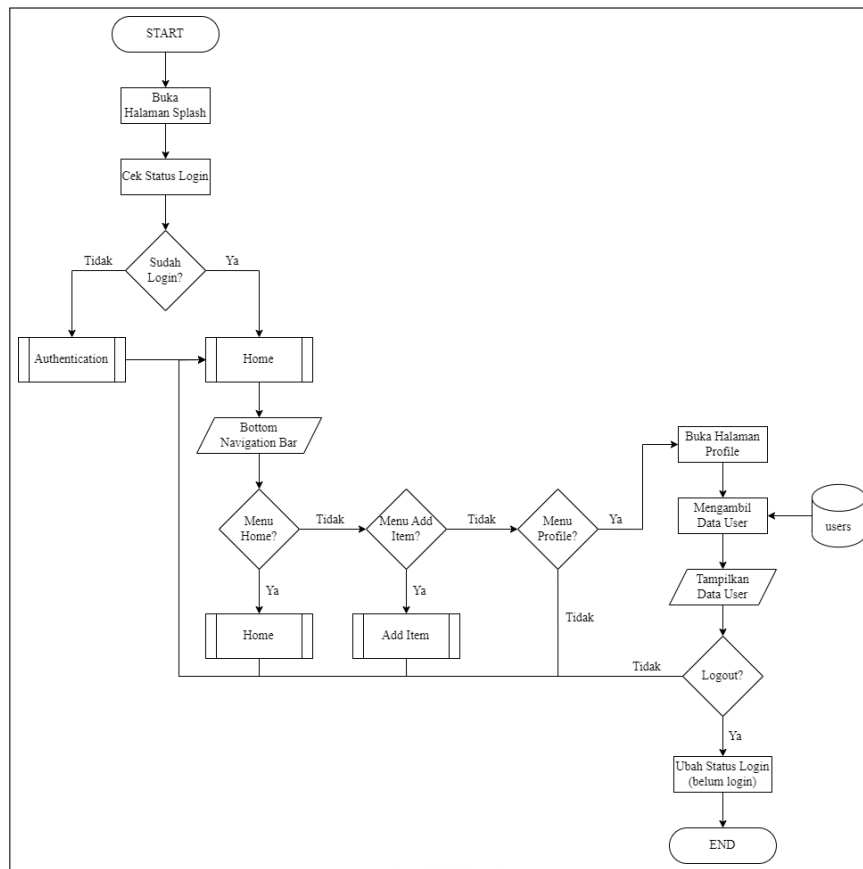
Gambar 3.2. Dokumentasi Studi Literatur

3.2 Design

Tahapan ini berisikan aktivitas yang berhubungan dengan perancangan alur aplikasi, pembuatan model aplikasi, dan pembuatan *mockup* aplikasi, mulai dari penentuan *color palette* hingga desain dan penempatan UI.

3.2.1 Alur Aplikasi

Alur aplikasi yang telah dirancang dapat dilihat melalui *flowchart* pada Gambar 3.3 di bawah ini.

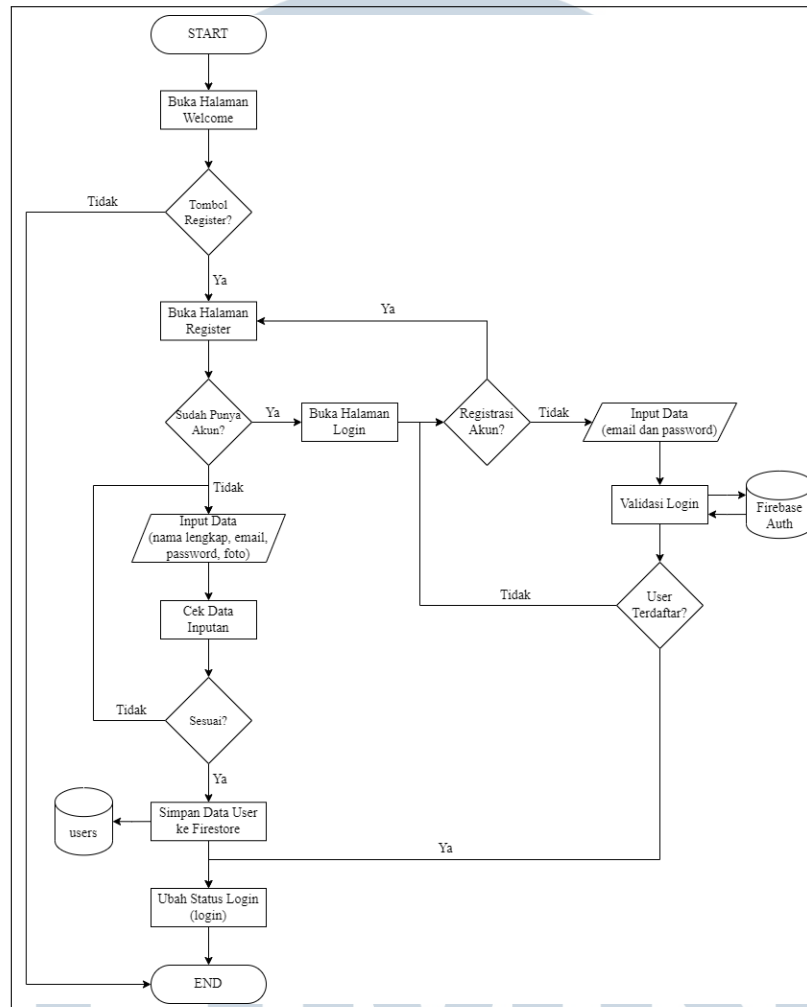


Gambar 3.3. Flowchart Alur Utama Aplikasi

Berdasarkan *flowchart* pada Gambar 3.3, aplikasi dimulai dengan membuka halaman *Splash* dan dilanjutkan dengan melakukan pengecekan status *login* (apakah *user* sudah *login* sebelumnya pada aplikasi atau belum). Jika *user* belum *login*, maka akan dilakukan *Authentication*. Sebaliknya, jika *user* telah *login*, maka akan dibuka halaman *Home*. Pada halaman ini terdapat 3 (tiga) *bottom navigation bar* yaitu Menu *Home*, Menu *Add Item*, dan Menu *Profile*.

Pada Menu *Profile*, dimulai dengan membuka halaman *Profile*. Selanjutnya aplikasi akan mengambil data *current user* (*user* yang sedang *login*) dari database *users*. Data yang telah diambil tersebut akan digunakan untuk menampilkan informasi (data) dari *user* berupa foto profil, nama lengkap, dan alamat email. Pada halaman ini, *user* dapat melakukan *logout* (keluar dari akun) dengan menekan tombol *logout*. Setelah tombol ditekan maka aplikasi akan mengubah status *login* yang sebelumnya *login* menjadi belum *login* dan membuat *user* keluar dari akunnya (*logout*).

A Authentication

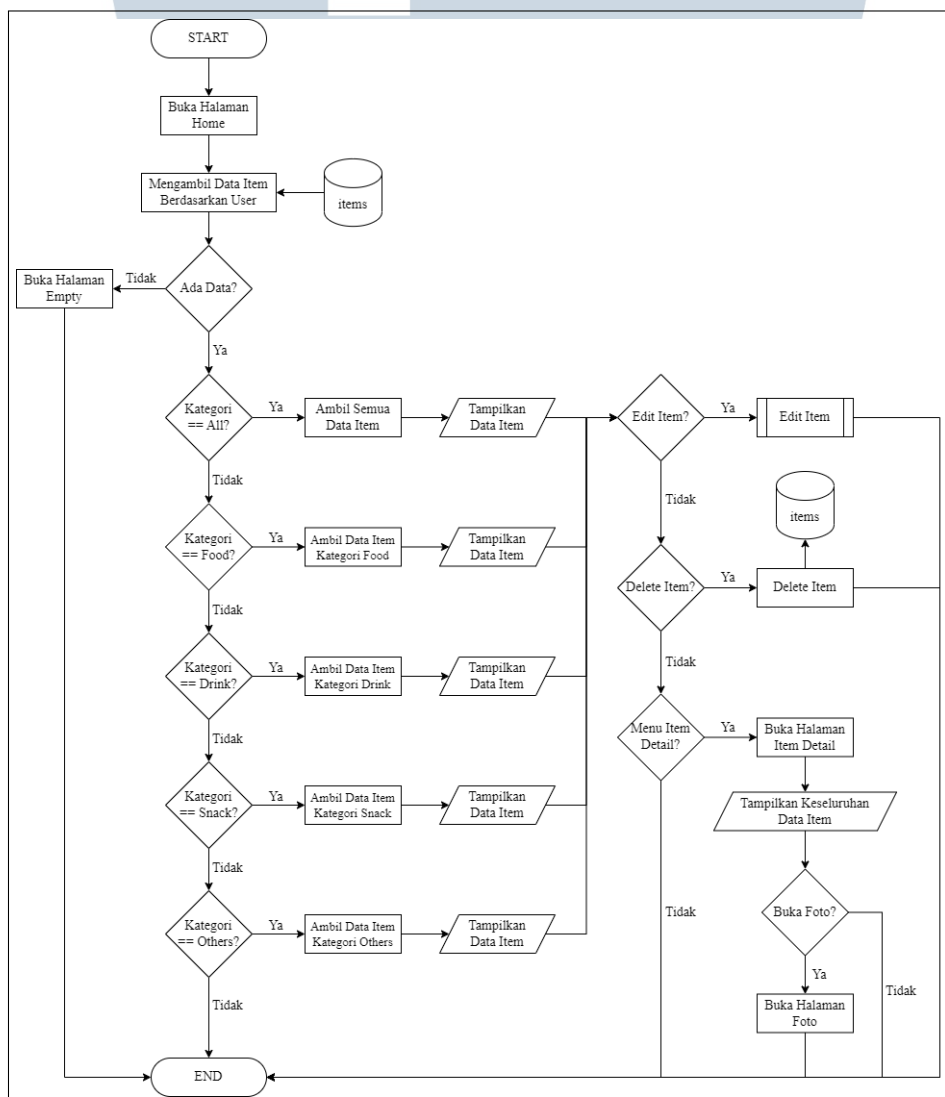


Gambar 3.4. *Flowchart Authentication*

Gambar 3.4 merupakan *flowchart* dari *Authentication* yang dimulai dengan membuka halaman *Welcome*. Pada halaman ini terdapat satu tombol *Register* yang digunakan untuk membuka halaman *Register*. Apabila *user* ingin melakukan pendaftaran akun maka *user* dapat melanjutkan berada pada halaman ini dan melakukan penginputan data seperti nama lengkap, alamat email, kata sandi, dan foto profil. Setelah itu, akan dilakukan pengecekan kesesuaian data dengan ketentuan yang ada, jika data yang dimasukkan aman maka data tersebut akan disimpan ke dalam *database users* dan registrasi akun telah selesai dilakukan. Sebaliknya, jika terdapat data yang belum sesuai maka *user* dapat melakukan penginputan data ulang.

Apabila *user* telah memiliki akun pada aplikasi dan ingin melakukan *login*, maka *user* dapat berpindah halaman ke *Login* dan memasukkan data berupa alamat email dan kata sandi. Kemudian aplikasi akan melakukan pengecekan terhadap data yang diterima dengan data pada Firebase Auth. Jika data yang diterima sesuai dengan salah satu data pada Firebase Auth (*user* terdaftar), maka *user* telah berhasil melakukan *login*. Setelah *user* berhasil melakukan registrasi akun atau *login*, aplikasi akan mengubah status *login* yang sebelumnya belum *login* menjadi *login*.

B Home



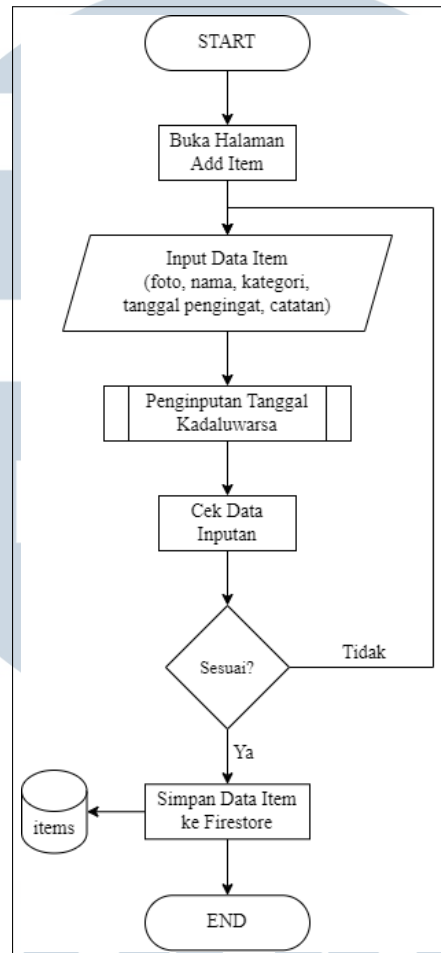
Gambar 3.5. Flowchart Home

Gambar 3.5 merupakan *flowchart* dari *Home* yang dimulai dengan membuka halaman *Home*. Setelah itu, aplikasi akan mengambil data *item* berdasarkan *current user* (*user* yang *login*) dari *database items*. Jika tidak terdapat data pada *database*, maka aplikasi akan membuka halaman *Empty*. Sebaliknya, jika terdapat data pada *database*, maka aplikasi akan menampilkan data berdasarkan kategori yang dipilih. Terdapat 5 (lima) kategori yang dapat dipilih yaitu *All* (Semua), *Food* (Makanan), *Drink* (Minuman), *Snack* (Makanan Ringan), dan *Others* (Lainnya).

Untuk setiap *item* yang ada dapat diubah (*edit*), dihapus, dan dibuka halaman *Item Detail*-nya. Halaman *Item Detail* digunakan untuk mendapatkan keseluruhan informasi yang tersimpan dari *item* yang dipilih. Selain itu, pada halaman ini, *user* juga dapat melihat foto produk dengan lebih jelas dan besar dengan cara menekan foto produk tersebut.



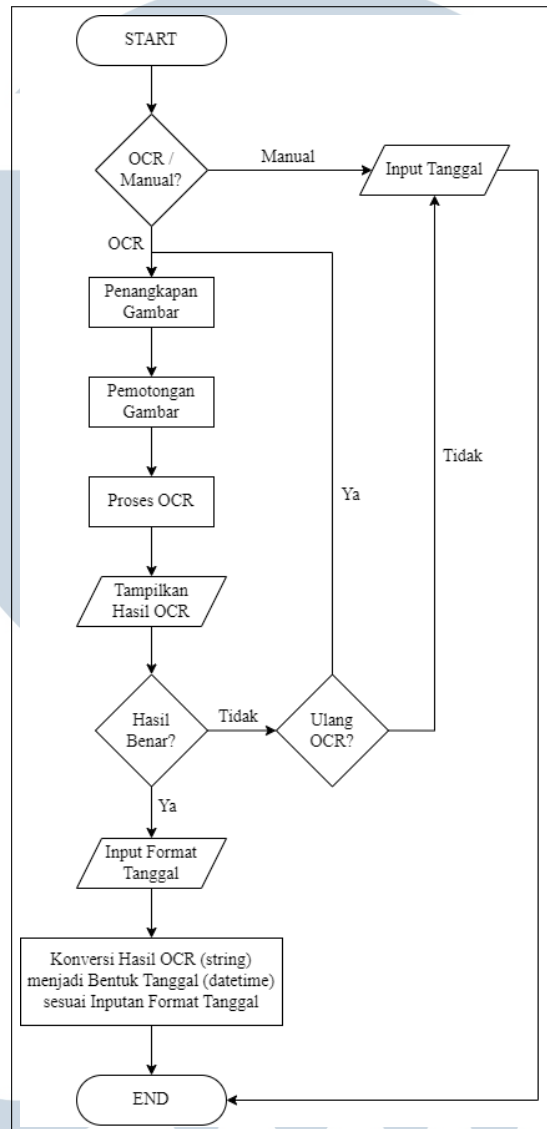
C Add Item



Gambar 3.6. Flowchart Add Item

Gambar 3.6 merupakan *flowchart* dari *Add Item* (penambahan *item*) yang dimulai dengan membuka halaman *Add Item*. Selanjutnya *user* dapat memasukkan data berupa foto, nama, tanggal kadaluwarsa (alur memasukkan tanggal kadaluwarsa akan dijelaskan pada subbab setelah ini), kategori, tanggal pengingat (*reminder*), dan catatan (*notes*). Kemudian dilakukan pengecekan terhadap data inputan apakah telah sesuai dengan ketentuan yang ada atau belum. Apabila inputan telah sesuai maka data akan disimpan ke dalam *database items*. Sebaliknya, jika inputan belum sesuai maka *user* dapat melakukan penginputan data ulang.

D Penginputan Tanggal Kadaluwarsa



Gambar 3.7. Flowchart Penginputan Tanggal Kadaluwarsa

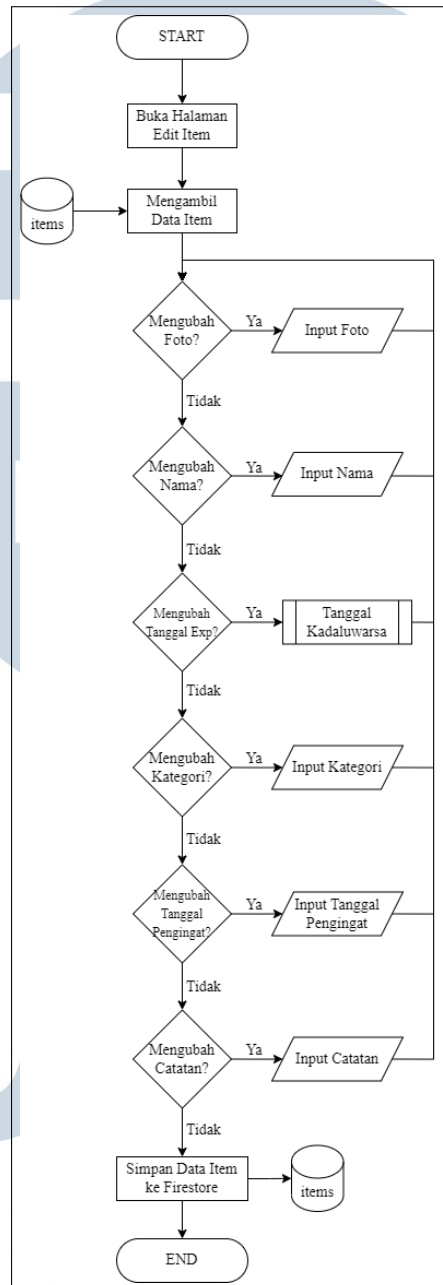
Gambar 3.7 merupakan *flowchart* dari alur penginputan tanggal kadaluwarsa yang menggunakan sistem OCR. Pertama-tama, aplikasi akan memunculkan *dialog box* untuk menanyakan apakah *user* ingin melakukan penginputan data tanggal kadaluwarsa melalui proses OCR atau manual. Jika *user* memilih menggunakan manual maka dapat langsung melakukan penginputan tanggal kadaluwarsa dengan menekan tanggal, bulan, dan tahun yang sesuai melalui *widget* kalender yang tersedia.

Sebaliknya, jika *user* memilih menggunakan proses OCR maka langkah awal yang akan dilakukan oleh aplikasi adalah membuka kamera dan menangkap gambar. Setelah itu, gambar hasil tangkapan akan mengalami proses pemotongan (*cropping*). Kemudian hasil *cropping* gambar tersebut akan mengalami proses pembacaan OCR. Setelah OCR berhasil membaca tulisan (tanggal kadaluwarsa) pada gambar, akan ditampilkan hasil pembacaan tersebut melalui *dialog box* dan dilakukan konfirmasi kepada *user* apakah hasil pembacaan tersebut telah benar atau belum.

Jika hasil tersebut sudah benar, maka *user* dapat memasukkan format tanggal dengan cara memilih salah satu dari 7 (tujuh) format tanggal yang tersedia pada *dropdown button*. Pemilihan format tanggal oleh *user* dikarenakan sampai saat ini setiap produk makanan dan minuman yang beredar belum memiliki satu format tanggal resmi. Setelah memilih format tanggal maka aplikasi akan melakukan konversi secara otomatis dari *string* hasil pembacaan OCR ke dalam bentuk tanggal *datetime*. Sebaliknya, jika hasil tersebut belum benar, maka *user* diberikan opsi untuk memilih mengulang proses OCR lagi atau tidak. Jika *user* bersedia maka akan dilakukan proses OCR ulang dan jika *user* tidak bersedia maka *user* dapat memasukkan tanggal kadaluwarsa secara manual.



E *Edit Item*



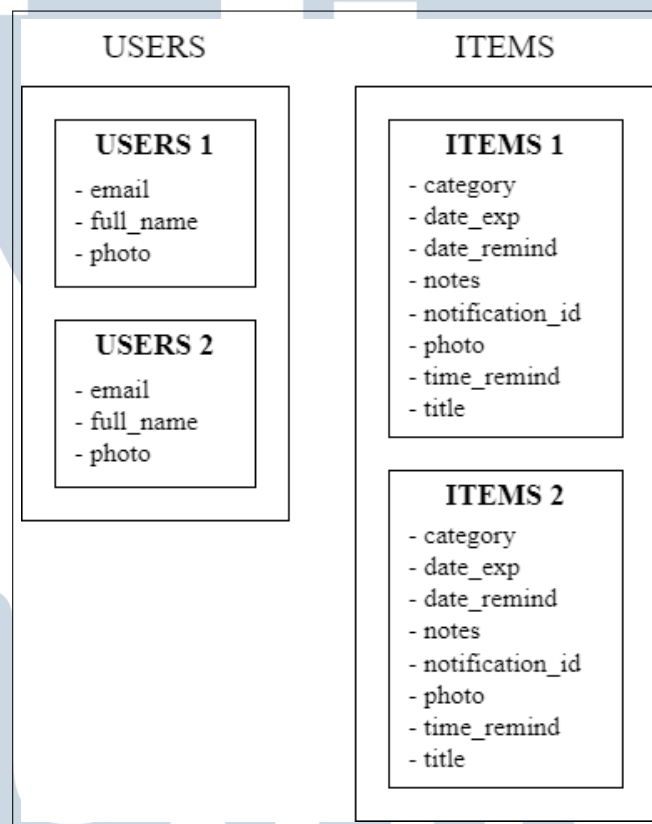
Gambar 3.8. *Flowchart Edit Item*

Gambar 3.8 merupakan *flowchart* dari *edit item* (pengeditan *item*) yang dimulai dengan membuka halaman *Edit Item*. Selanjutnya aplikasi akan mengambil data *item* sesuai dengan *item* yang dipilih untuk dimodifikasi dari *database items*. Adapun data yang dapat diubah adalah foto, nama, tanggal kadaluwarsa, kategori,

tanggal *reminder*, dan *notes*. Setelah *user* selesai melakukan input terhadap data yang ingin dimodifikasinya, aplikasi akan kembali menyimpan data-data baru yang baru saja dimasukkan oleh *user* ke dalam *database items*.

3.2.2 Model Aplikasi

Aplikasi ini menggunakan basis data *Firestore Database* (Firebase) dengan struktur yang digambarkan pada Gambar 3.9 di bawah ini.



Gambar 3.9. Struktur *Firestore Database*

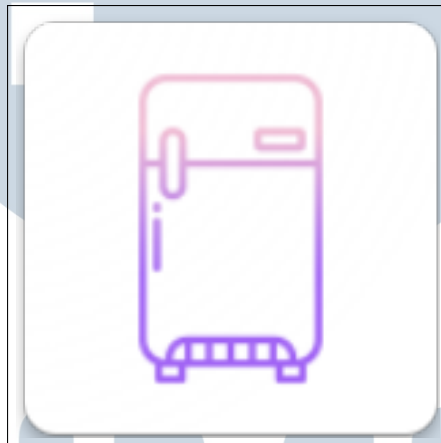
Berdasarkan struktur yang terdapat pada Gambar 3.9, terdapat 2 (dua) buah *collection* yang disimpan oleh Firebase yaitu *Users* dan *Items*. *Users* akan menyimpan data-data berupa email, nama lengkap, dan foto dari setiap user yang melakukan registrasi akun. Sedangkan *items* akan menyimpan data-data berupa kategori, tanggal kadaluwarsa, tanggal dan waktu *reminder*, *notes*, *notification id* (digunakan untuk menghapus tanggal *reminder* sebelumnya saat melakukan pengeditan tanggal *reminder* atau penghapusan *item*), foto produk, dan nama produk dari setiap *item* yang disimpan oleh *user*. Setiap *user* hanya dapat

mengakses *item* yang dimasukkan oleh *user* tersebut saja dengan memanfaatkan UID (*User ID*) yang terdapat pada *Firebase Authentication*.

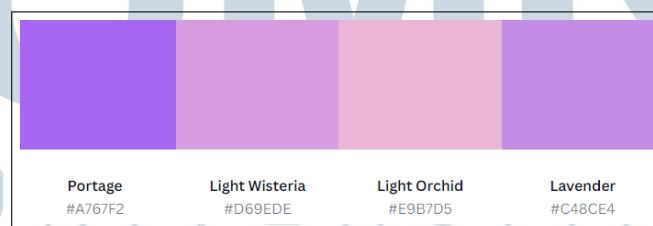
3.2.3 Mockup Aplikasi

A Nama dan Logo Aplikasi

Aplikasi ini bernama RYFABE (*Remind Your Foods And Beverages Expired*) yang memiliki arti sesuai dengan tujuan utama aplikasi ini dibangun untuk membantu *user* mengingat tanggal kadaluwarsa dari makanan dan minuman yang disimpan oleh *user*. Logo dari aplikasi ini dapat dilihat pada Gambar 3.10 beserta *color palette*-nya yang terdapat pada Gambar 3.11.



Gambar 3.10. Logo RYFABE





Gambar 3.11. Color Palette Logo RYFABE

Logo dari RYFABE sendiri menyerupai kulkas atau lemari penyimpanan yang biasanya digunakan untuk menyimpan makanan dan minuman. Dengan kata lain, RYFABE merupakan lemari penyimpanan data tanggal kadaluwarsa dari setiap *item* (makanan dan minuman) yang disimpan oleh *user*.

C Daftar Aset

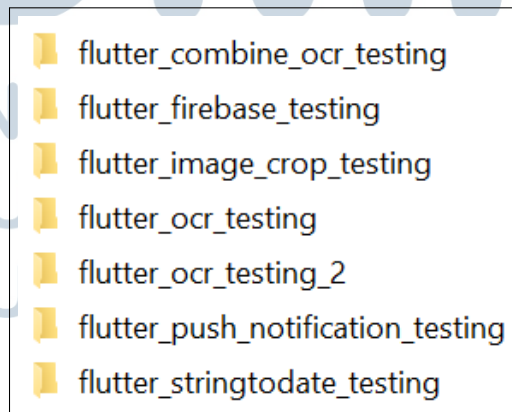
Tabel 3.1 di bawah ini merupakan daftar aset di luar UI atau *widget* dari Flutter yang digunakan pada aplikasi ini.

Tabel 3.1. Daftar Aset

Gambar	Penjelasan	Sumber
	Logo Aplikasi	www.icons8.com
	<i>Background Halaman Empty</i>	www.icons8.com

3.3 Development

Tahapan ini merupakan awal dari pengembangan produk, masih berupa potongan-potongan fitur yang berdiri secara independen, misalnya pembuatan halaman UI saja yang belum dapat berfungsi, pembuatan fitur notifikasi apabila telah mencapai batas tanggal dan waktu yang ditentukan, pembuatan pembacaan OCR dengan menggunakan kamera, dan potongan fitur lainnya dalam ranah pengembangan produk sesuai dengan *requirement* yang dimiliki. Gambar 3.14 di bawah ini merupakan dokumentasi pengerjaan pada tahapan ini.



Gambar 3.14. Dokumentasi *Development Process*

3.4 Implement

Tahapan ini merupakan tahapan lanjutan dari tahapan *development* yang dimana pada tahapan ini akan menggabungkan keseluruhan pengembangan produk yang telah dikembangkan sebelumnya untuk dijadikan sebagai suatu aplikasi yang utuh. Selain itu, tahapan ini juga akan mengembangkan fungsi-fungsi lanjutan lainnya dari produk. Hasil dari tahapan ini akan dibahas pada Bab 4.

3.5 Evaluation

Tahapan ini dilakukan untuk melakukan pengujian dan evaluasi terhadap aplikasi yang telah dibangun. Pengujian terdiri dari pengujian tingkat akurasi OCR dan lama waktu pembacaan OCR. Pengujian tingkat akurasi OCR dilakukan pada 2 (dua) buah *packages* OCR yang memiliki sifat yang berbeda (*offline* dan *online*) untuk dibandingkan hasil akurasi antar *packages* tersebut. Kedua *packages* akan diuji dengan menggunakan 22 sampel foto tanggal kadaluwarsa dari berbagai produk menggunakan perhitungan CER. Pengujian lama waktu pembacaan OCR akan dilakukan dengan menggunakan sampel foto tanggal kadaluwarsa yang sama melalui potongan kode pada aplikasi. Untuk pengujian terhadap lama waktu pembacaan OCR hanya dilakukan pada *packages* yang digunakan pada aplikasi. Hasil dari tahapan ini juga akan dibahas pada Bab 4.

3.6 Report Finalization

Tahapan ini merupakan tahapan akhir dalam penelitian ini yang berisikan finalisasi dan publikasi laporan.

U M M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A