Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Perpustakaan Univeristas Multimedia Nusantara

BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Beberapa metode penelitian yang digunakan dalam keperluan penelitian ini:

1. Studi Literatur

   Pencarian dan pengumpulan informasi mengenai penjelasan dan konsep mengenai metode beserta kelebihannya dibandingkan metode lainnya dan bagaimana cara kerjanya dari berbagai referensi yang tersedia di internet.

2. Pengumpulan Data

   Data yang akan digunakan dalam penelitian dalam hal ini adalah resep masakan. Data resep tersebut diambil dari *selerasa.com*.

3. Perancangan

   Ada beberapa perancangan yang akan dibuat dalam penelitian ini:

   a. Perancangan antar muka aplikasi.

   b. Perancangan sistem arsitektur.

   c. Pembuatan alur kerja dari sistem dengan *flowchart*.

   d. Pembuatan *modeling* data dengan Conceptual Data Model.

4. Implementasi

   Implementasi yang dilakukan meliputi implementasi antarmuka dan algoritma. Implementasi dilakukan pada aplikasi resep masakan dengan *platform mobile* Android dan dikembangkan menggunakan *framework Ionic*.

5. Pengujian

   Pengujian apakah implementasi algoritma yang digunakan berjalan dengan baik dengan penggunaan aplikasi oleh *user* untuk kemudian dievaluasi.

6. Evaluasi

   Pengevaluasian dengan cara melihat tingkat *acceptance* aplikasi dengan menggunakan *Technology Acceptance Model*. Data yang digunakan dalam evaluasi didapatkan dari survei yang akan diisi oleh 60 responden dengan *random sampling* berdasarkan penilaian pengguna terhadap aplikasi terkait.

3.2 Perancangan Aplikasi

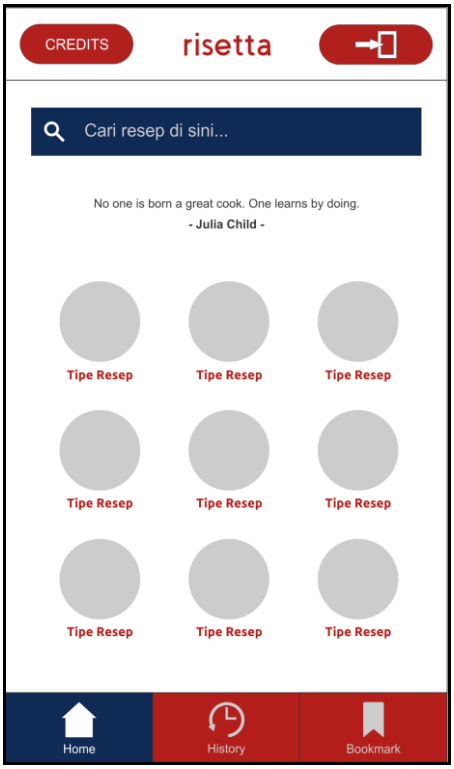
Perancangan aplikasi yang dilakukan meliputi perancangan antarmuka aplikasi, perancangan arsitektur sistem, *flowchart*, dan membuatan *modeling* data dengan *Conceptual Data Model*.

3.2.1 Perancangan Antarmuka

Pada proses perancangan dilakukan juga perancangan antarmuka untuk aplikasi Risetta. Perancangan desain *page*, *button*, logo dibuat menggunakan *CorelDRAW*. Halaman yang dirancang antara lain halaman Home, Pencarian, Daftar Resep, Detail Resep, History, dan Bookmark.

a. Halaman Home

Setelah animasi *splash screen* telah selesai, maka pengguna diarahkan ke halaman Home. Pada halaman ini berisi *input field search* untuk mencari resep yang ingin dicari dan ada sembilan menu tipe resep yang berbeda-beda yang ketika ditekan akan mengarahkan pengguna ke halaman Daftar Resep (Gambar 3.2).



Gambar 3.2 Rancangan Antarmuka Halaman Home

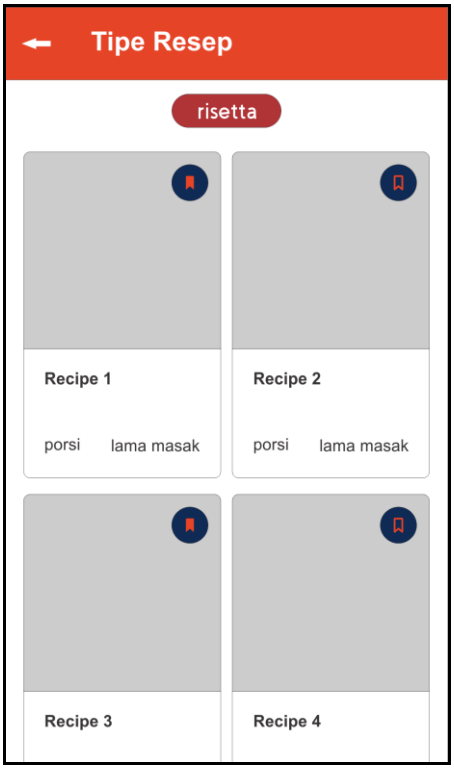
b. Daftar Pencarian

Pada halaman Home, terdapat *input field* untuk dilakukan pencarian resep yang ingin dicari oleh pengguna. Di sini pengguna dapat mengetik secara langsung resep yang ingin dicari, dan daftar pencarian akan di-*update* secara otomatis, jika pengguna menekan salah satu dari daftar tersebut maka diarahkan langsung ke halaman Resep Detail. Apabila resep tidak ditemukan maka tetap dapat dilakukan pencarian untuk menghitung tingkat kemiripan *query* yang dicari dengan daftar resep yang dimiliki menggunakan algoritma Jaro-Winkler Distance. Tanda panah "←" untuk kembali ke halaman Home, dan tanda "x" untuk menghapus semua karakter yang telah diketik. Halaman ini juga terdapat tombol "Credits" dan keluar aplikasi (Gambar 3.3).

Gambar 3.3 Rancangan Antarmuka Pencarian Resep

c. Halaman Daftar Resep

Pada halaman ini (Gambar 3.4) ditampilkan semua resep yang termasuk dalam kelompok tipe resep yang dipilih pada halaman Home. Pada halaman ini, resep dibagi-bagi menjadi beberapa *card*. Pada card tersebut, di bagian kanan atas terdapat tombol berbentuk lingkaran dengan *icon bookmark* yang bila ditekan akan memasukkan menu tersebut ke dalam daftar di halaman Bookmark serta menampilkan *toast* baik ditambahkan dan dihapus dari Bookmark apabila status resep tersebut adalah *bookmarked* sebelumnya (Gambar 3.5).



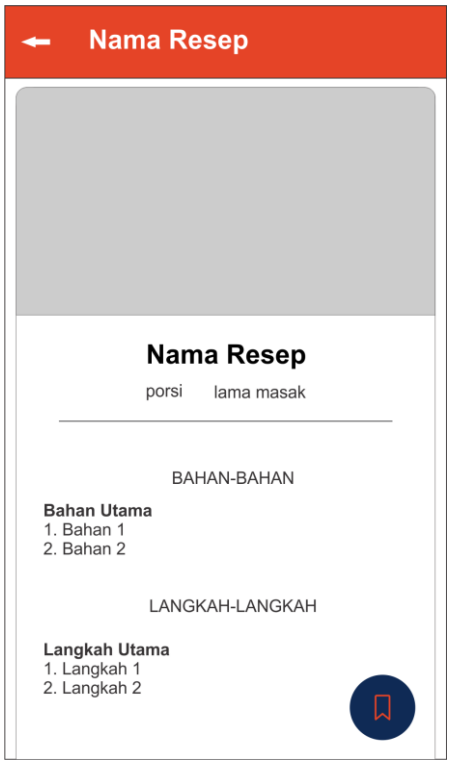
Gambar 3.4 Rancangan Antarmuka Halaman Daftar Resep

Gambar 3.5 Rancangan Antarmuka Toast Menambahkan dan Menghapus Bookmark

d. Halaman Detail Resep

Halaman Resep Detail adalah halaman yang dituju apabila pengguna menekan pada salah satu daftar resep di *input field search*, daftar resep, dan halaman Bookmark. Pada halaman ini pengguna akan disajikan informasi detail dari resep yang dipilih, mulai dari gambar, bahan-bahan, langkah-langkah, dan lain sebagainya. Pada halaman ini juga terdapat *floating button bookmark* yang cara

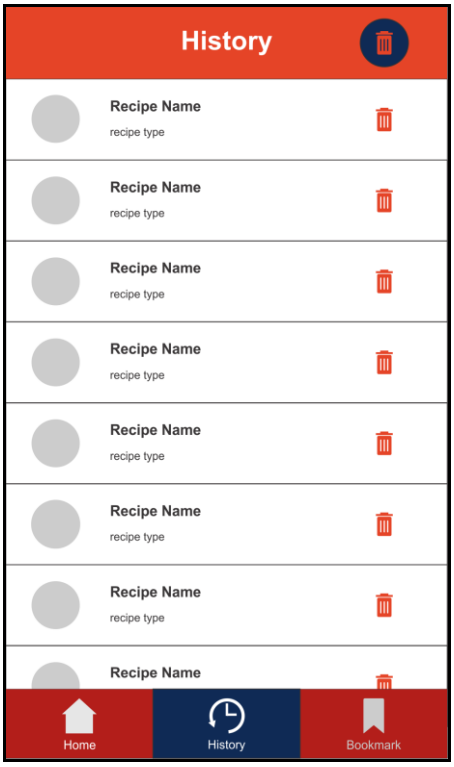
kerjanya sama pada halaman Daftar Resep, yaitu menambah resep ke halaman Bookmark maupun menghapus resep dari halaman Bookmark (Gambar 3.6).



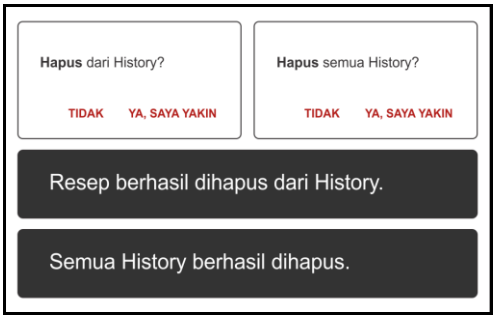
Gambar 3.6 Rancangan Antarmuka Halaman Resep Detail

e. Halaman History

Pada halaman ini (Gambar 3.7) ditampilkan daftar resep yang pernah dikunjungi oleh pengguna. Pengguna juga dapat menghapus salah satu maupun semua resep dari daftar History. Apabila pengguna ingin menghapus semua *history* maka muncul *alert* untuk memvalidasi bahwa pengguna yakin untuk menghapus dan juga *toast* ketika berhasil dihapus (Gambar 3.8).



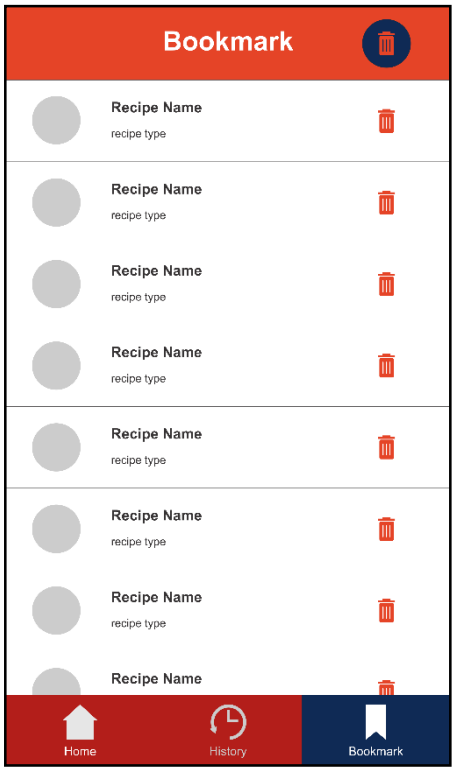
Gambar 3.7 Rancangan Antarmuka Halaman History



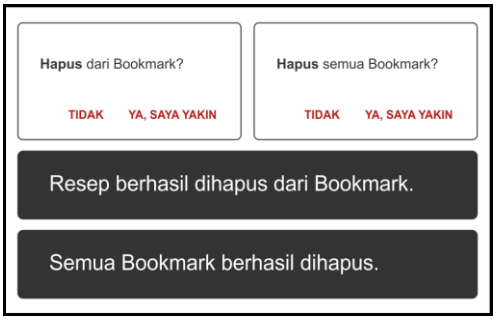
Gambar 3.8 Alert dan Toast Hapus History

f. Halaman Bookmark

Pada halaman ini (Gambar 3.9) ditampilkan daftar resep yang pernah di-*bookmark* oleh pengguna. Pengguna juga dapat menghapus salah satu maupun semua resep dari daftar Bookmark. Apabila pengguna ingin menghapus semua *bookmark* maka muncul *alert* untuk memvalidasi bahwa pengguna yakin untuk menghapus dan juga *toast* ketika berhasil dihapus (Gambar 3.10).



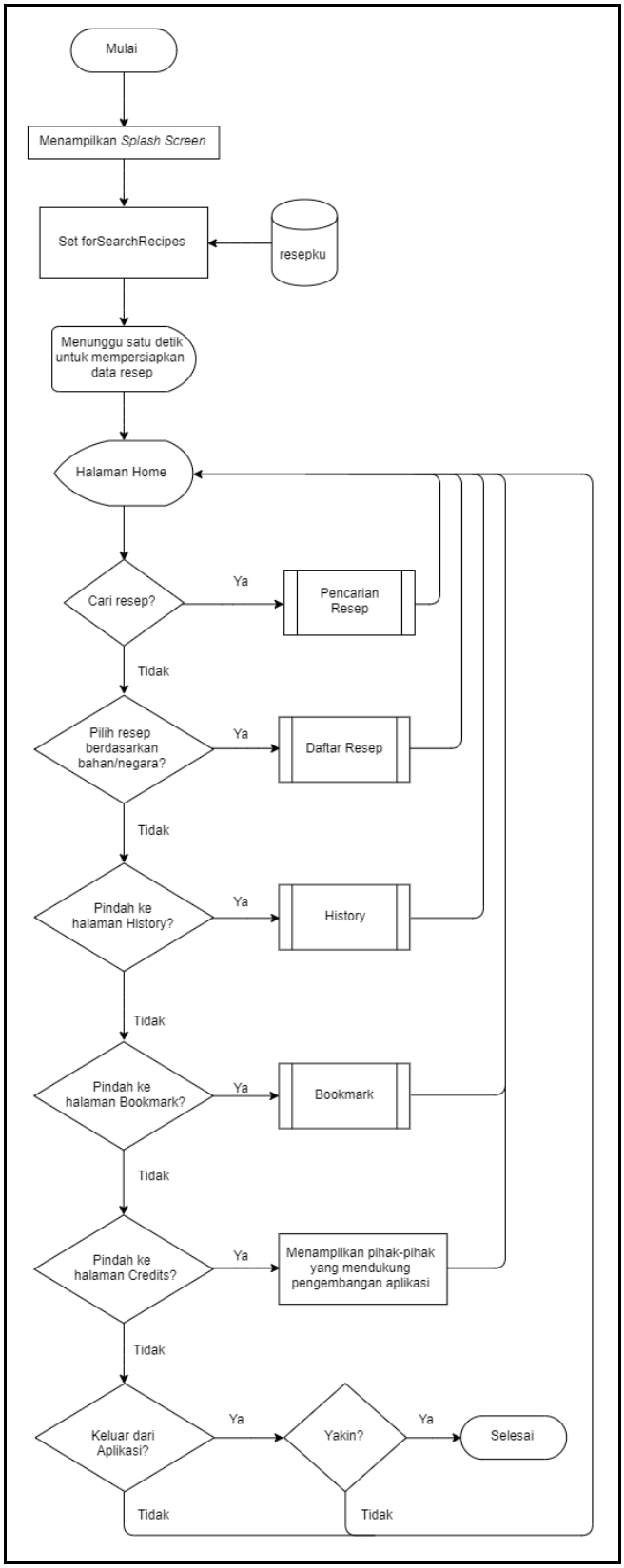
Gambar 3.9 Rancangan Antarmuka Halaman Bookmark



Gambar 3.10 Alert dan Toast Hapus Bookmark

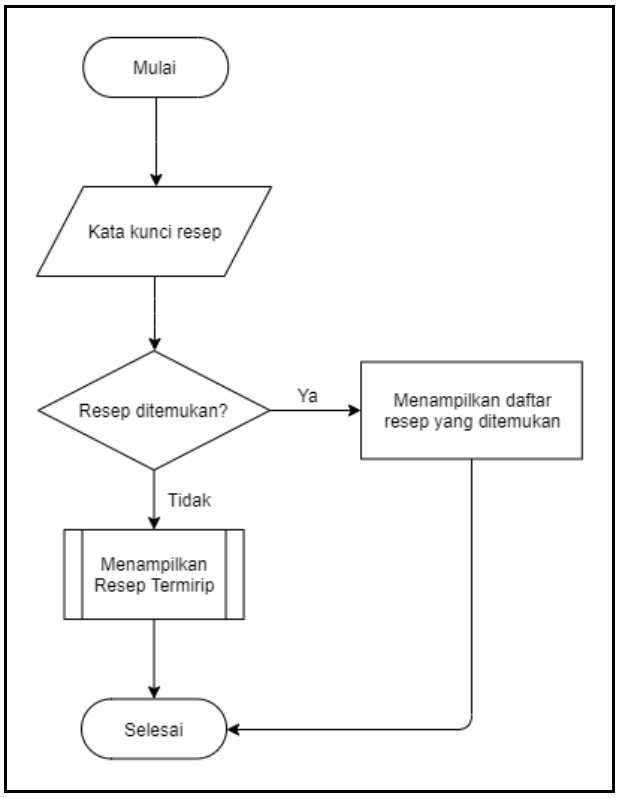
3.2.2 Flowchart

Flowchart menjelaskan alur kerja dari aplikasi Risetta. Gambar 3.11 merupakan Flowchart utama pada aplikasi. Pada awalnya aplikasi akan menampilkan *splash screen*, ketika animasi *splash screen* telah selesai maka pengguna menunggu selama satu detik guna mempersiapkan datar resep pada fitur pencarian kemudian baru diarahkan ke halaman Home. Pada halaman ini pengguna dapat langsung mencari resep yang ingin dicari dengan kata kunci yang diinginkan. Halaman utama ini dapat melakukan navigasi ke halaman Daftar Resep, History, Bookmark, Credits, serta dapat keluar dari aplikasi.



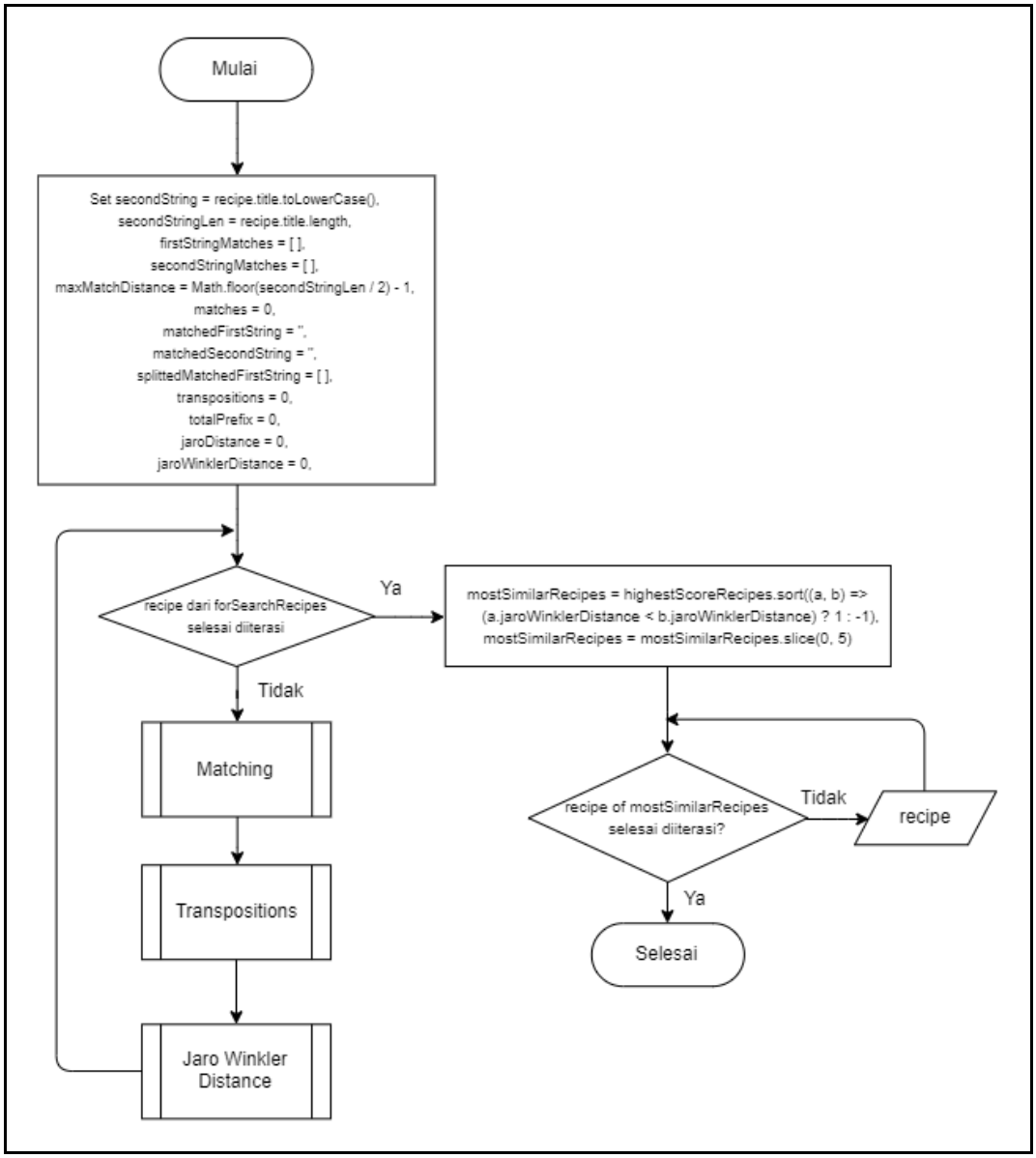
Gambar 3.11 Flowchart Aplikasi Risetta

Gambar 3.12 merupakan Flowchart Pencarian Kata, dari kata kunci yang dimasukkan oleh pengguna, apabila ditemukan kata yang sesuai dengan data yang ada dalam database akan ditampilkan, bila tidak maka akan dihitung kemiripannya dengan menggunakan algoritma Jaro-Winkler Distance.



Gambar 3.12 Flowchart Pencarian Resep

Gambar 3.13 menjelaskan Flowchart untuk menampilkan lima resep termirip. Pada *flowchart* ini akan mengiterasi *array* forSearchRecipes yang telah didapatkan dari *storage*. Selama iterasi akan melakukan proses mendapatkan variabel-variabel yang digunakan untuk perhitungan Jaro-Winkler Distance. Proses dibagi menjadi beberapa modul yaitu Matching, Transpositions, dan Jaro-Winkler Distance. Setelah selesai diiterasi maka highestScoreRecipes akan diurutkan dari yang memiliki nilai tertinggi hingga terendah dan diambil lima teratas.

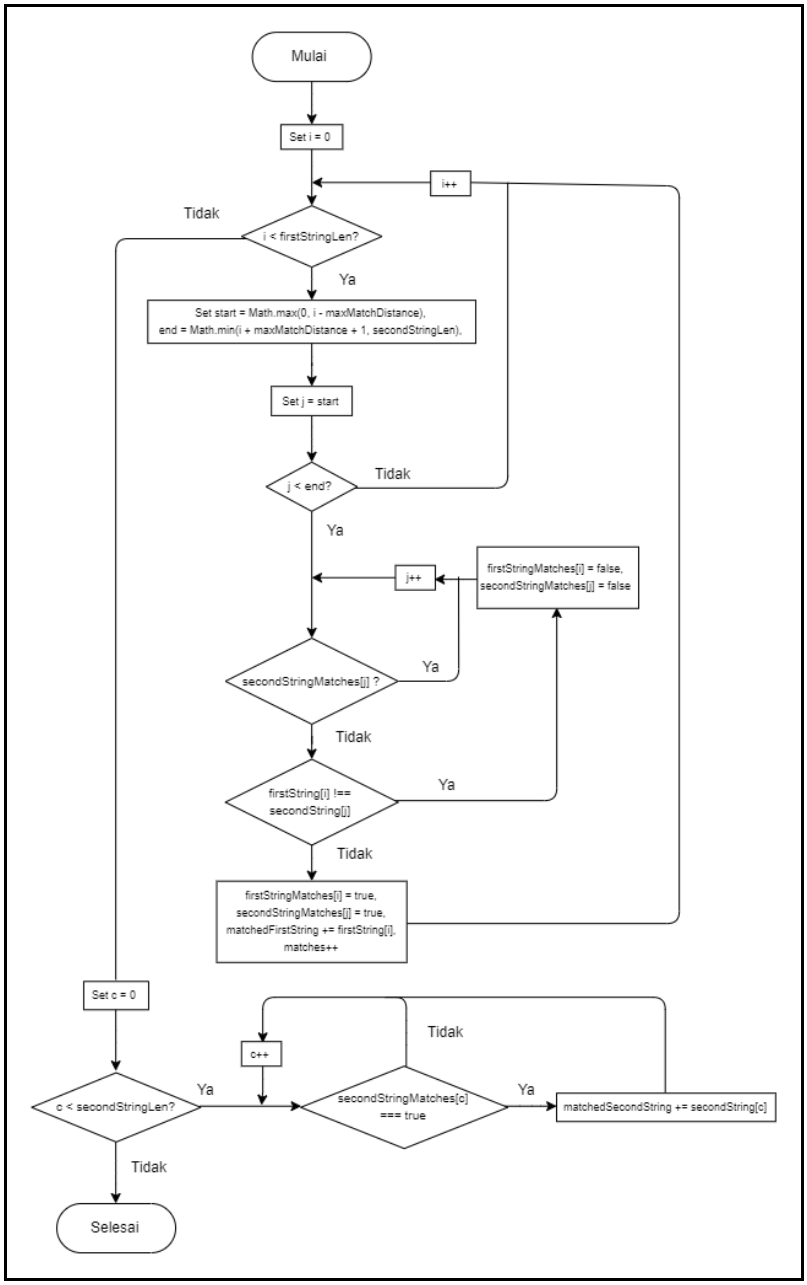


Gambar 3.13 Flowchart Menampilkan Resep Termirip

Gambar 3.14 merupakan Flowchart Matching menjelaskan bagaimana karakter dari *string* pertama yang berasal dari pengguna dan karakter *string* kedua yang berasal dari *database* dibandingkan untuk menandai posisi karakter dalam kedua *string* yang memiliki pasangan untuk proses perhitungan transposisi dan mendapatkan nilai *match*. Untuk mencari karakter apa saja yang *match* antara satu *string* dengan *string* yang lainnya, perlu dicari nilai *start* dan *end* untuk dijadikan

batasan perbandingan. Setiap karakter pada string pertama akan dibandingkan satu per satu dengan string dua dengan batasan nilai *start* dan *end* yang telah ditentukan sebelumnya. Apabila terdapat kesamaan antar dua karakter maka akan diberi penanda atau *flag* dengan nilai adalah *true* dan nilai *match* akan bertambah, jika tidak akan diberikan nilai False. Kemudian akan dilanjutkan ke karakter *index* berikutnya pada *string* pertama untuk mencari pasangan lagi pada *string* kedua.

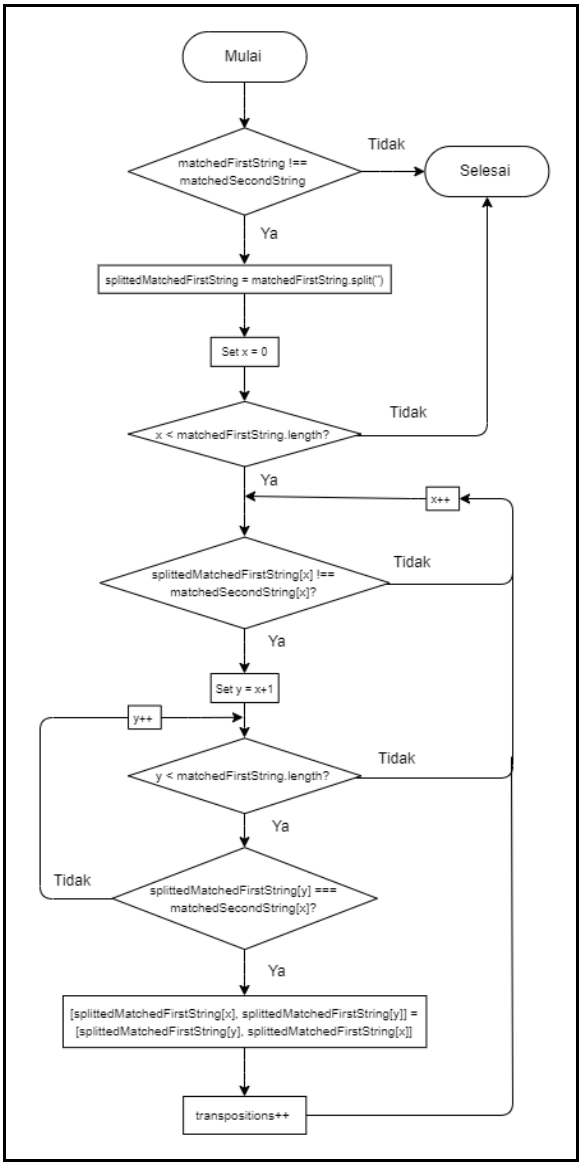
Karakter yang sudah *match* pada *string* kedua akan diabaikan, sehingga langsung dilanjutkan ke *index* berikutnya pada *string* kedua. Proses ini dilakukan terus menerus hingga semua karakter pada string pertama telah selesai dibandingkan dengan *string* kedua. Setelah mendapatkan nilai *match* maka dilakukan penggabungan karakter untuk string kedua yang *match* untuk kemudian digunakan pada proses perhitungan transposisi dengan cara mengecek satu per satu nilai *true* pada variabel string kedua yang *match*. Jika *true* maka dilakukan penggabungan karakter, jika *false* maka tidak melakukan proses apapun. Proses ini dilakukan hingga berhasil mengecek semua nilai karakter.



Gambar 3.14 Flowchart Matching

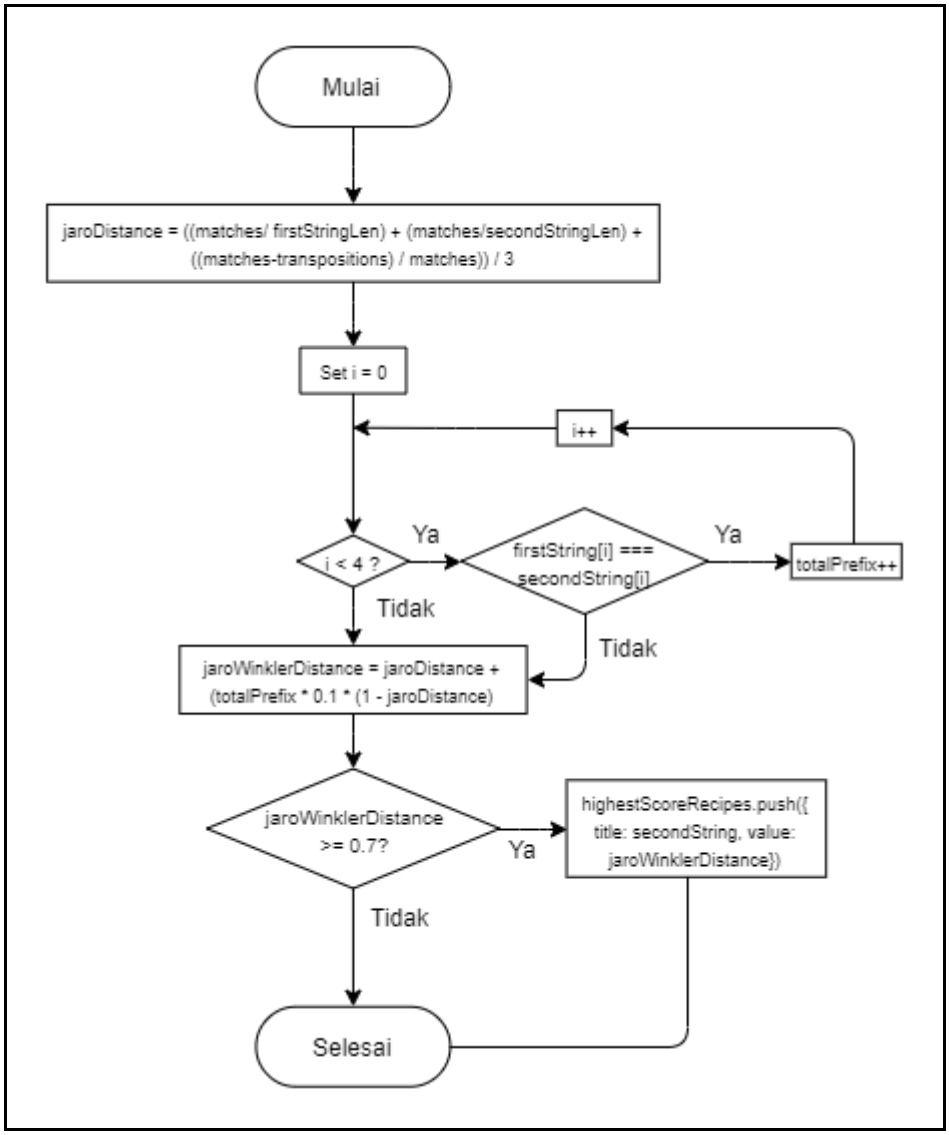
Gambar 3.15 merupakan Flowchart Transpositions. Flowchart ini menjelaskan cara untuk mendapatkan nilai transposisi. Nilai transposisi dihitung berdasarkan karakter-karakter yang berpasangan antara dua string dan dihitung hingga kedua string yang berpasangan tersebut sama persis satu sama lain. Kedua *string* yang berisi karakter apa saja yang *match* antara satu sama lain akan

dibandingkan. Jika terdapat ketidakcocokan urutan antara *string* pertama dan kedua maka akan dilakukan perhitungan transposisi. Kedua *string* tersebut akan dipisahkan per karakter kemudian dibandingkan dan dicari pada *index* atau posisi mana kedua karakter itu tertukar. Jika didapatkan maka akan langsung dilakukan proses pertukaran dan nilai transposisi bertambah dan langsung mencari lagi pada *index* berikutnya. Proses dilakukan terus-menerus hingga *string match* yang pertama dan kedua adalah sama persis.



Gambar 3.15 Flowchart Transpositions

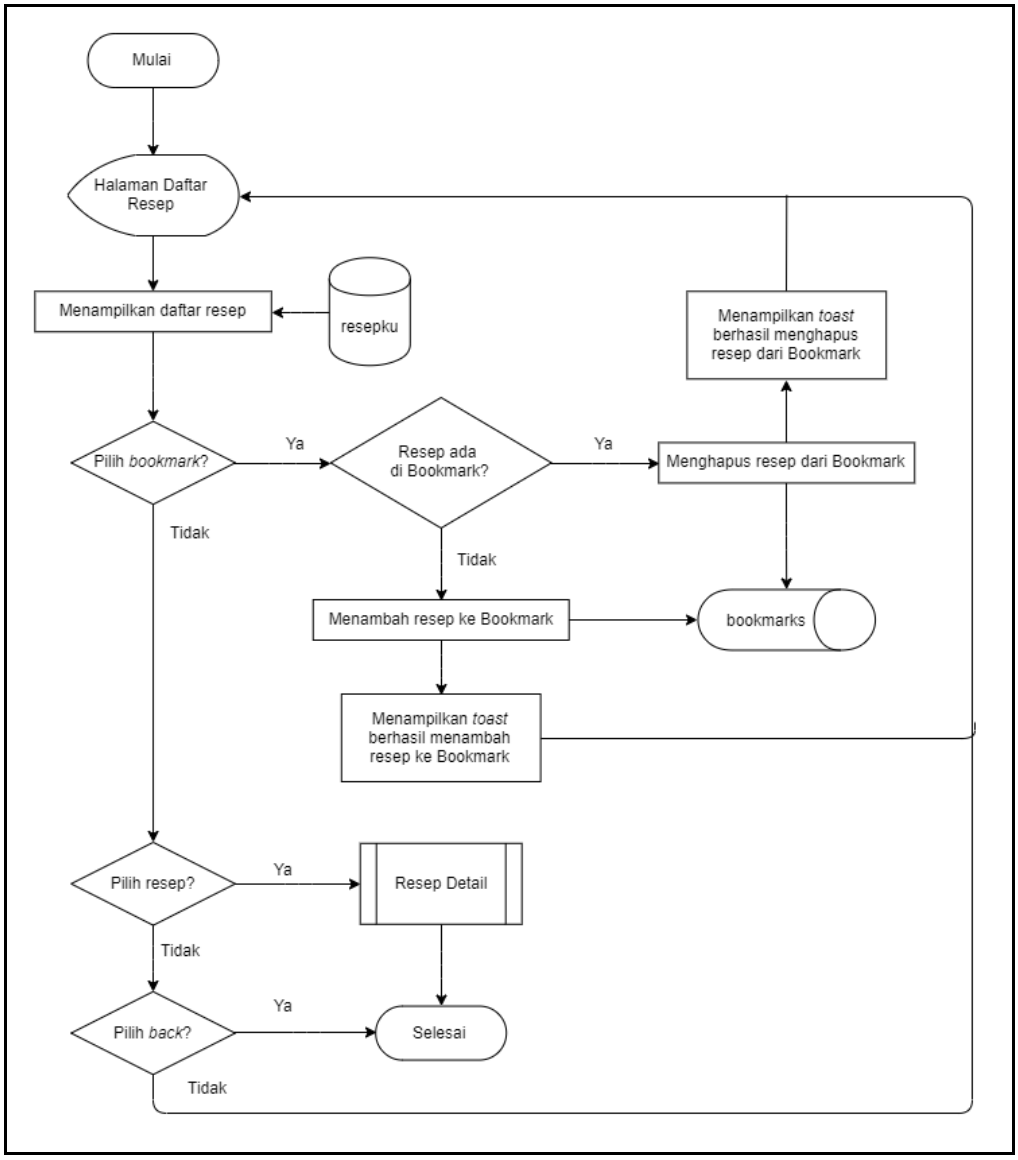
Gambar 3.16 menjelaskan perhitungan dari Jaro-Winkler Distance. Untuk mendapatkan nilai tersebut, perlu mendapatkan nilai Jaro Distance terlebih dahulu dan menghitung total prefiks yang sama antara dua string yang dibandingkan dengan nilai maksimal adalah empat. Nila Jaro-Winkler yang sudah didapatkan akan dimasukkan ke dalam highestScoreRecipes dengan kondisi nilai >= 0.7.



Gambar 3.16 Flowchart Jaro-Winkler Distance

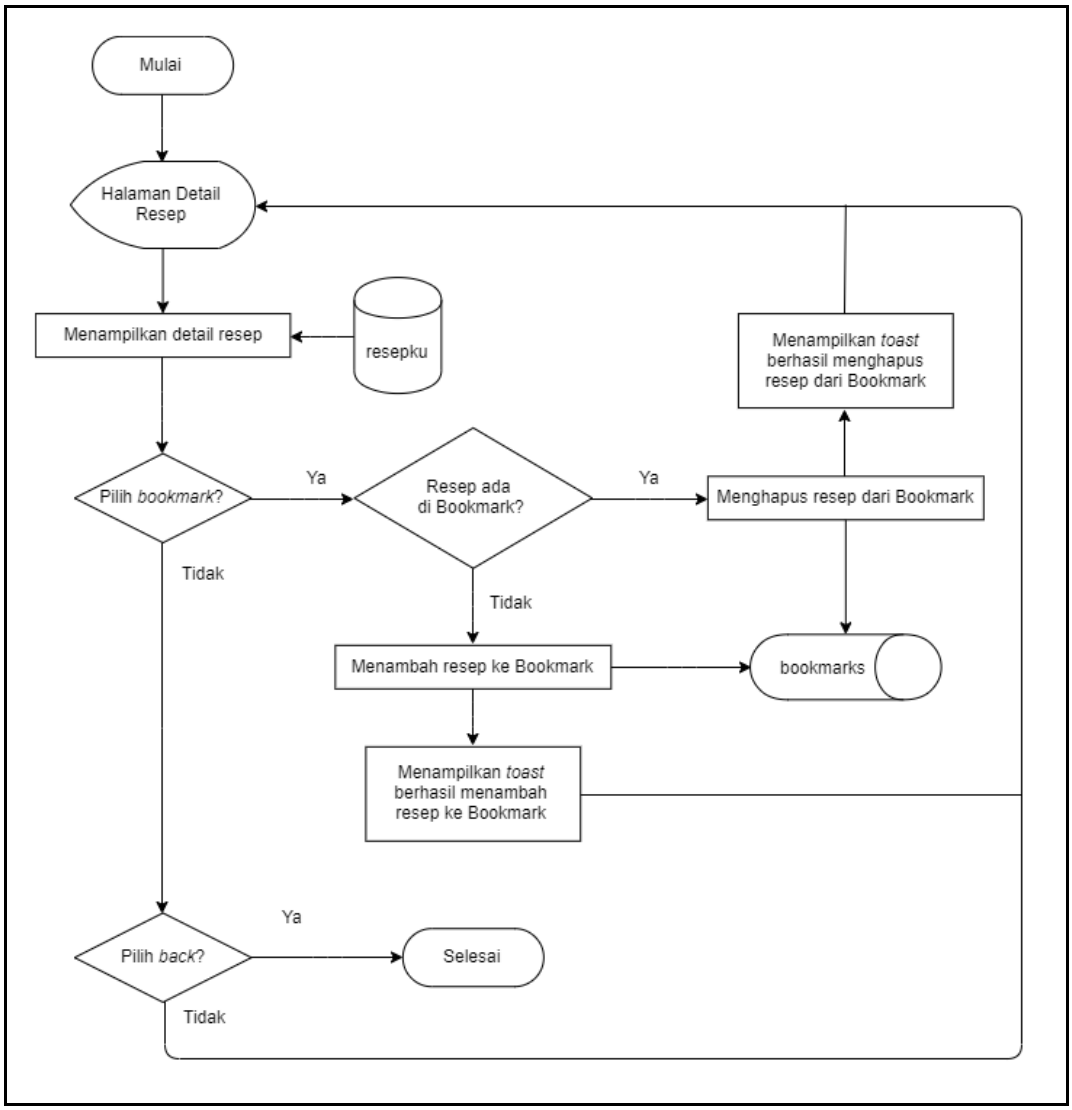
Gambar 3.17 merupakan Flowchart untuk halaman Daftar Resep. Pada halaman ini, ditampilkan daftar resep sesuai dengan menu yang dipilih pada

halaman beranda. Pada halaman ini terdapat pilihan untuk menyimpan resep ke dalam Bookmark. Jika status pada resep terkait pada Bookmark adalah belum *bookmarked*, maka resep akan dimasukan ke dalam Bookmark, sebaliknya jika dalam status *bookmarked* maka akan dihapus dan akan menampilkan *toast* pada *event* terkait. Jika memilih salah satu menu pada halaman ini maka akan diarahkan ke halaman Detail Resep.



Gambar 3.17 Flowchart Daftar Resep

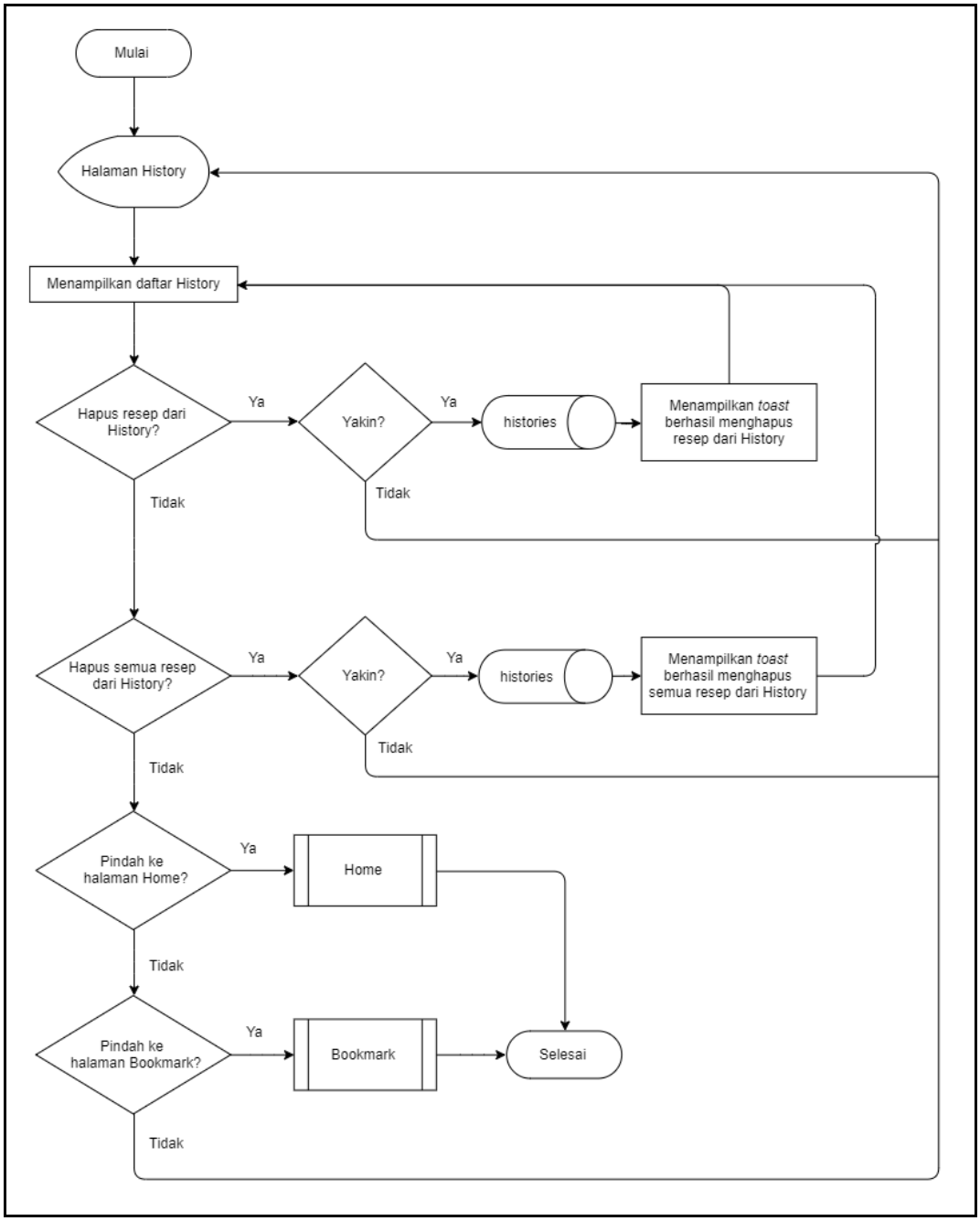
Gambar 3.18 merupakan Flowchart untuk halaman Detail Resep. Pada halaman ini, ditampilkan detail dari resep yang dipilih dalam halaman Daftar Resep. Di sini juga terdapat pilihan untuk menyimpan maupun menghapus resep terkait dari Bookmark.



Gambar 3.18 Flowchart Detail Resep

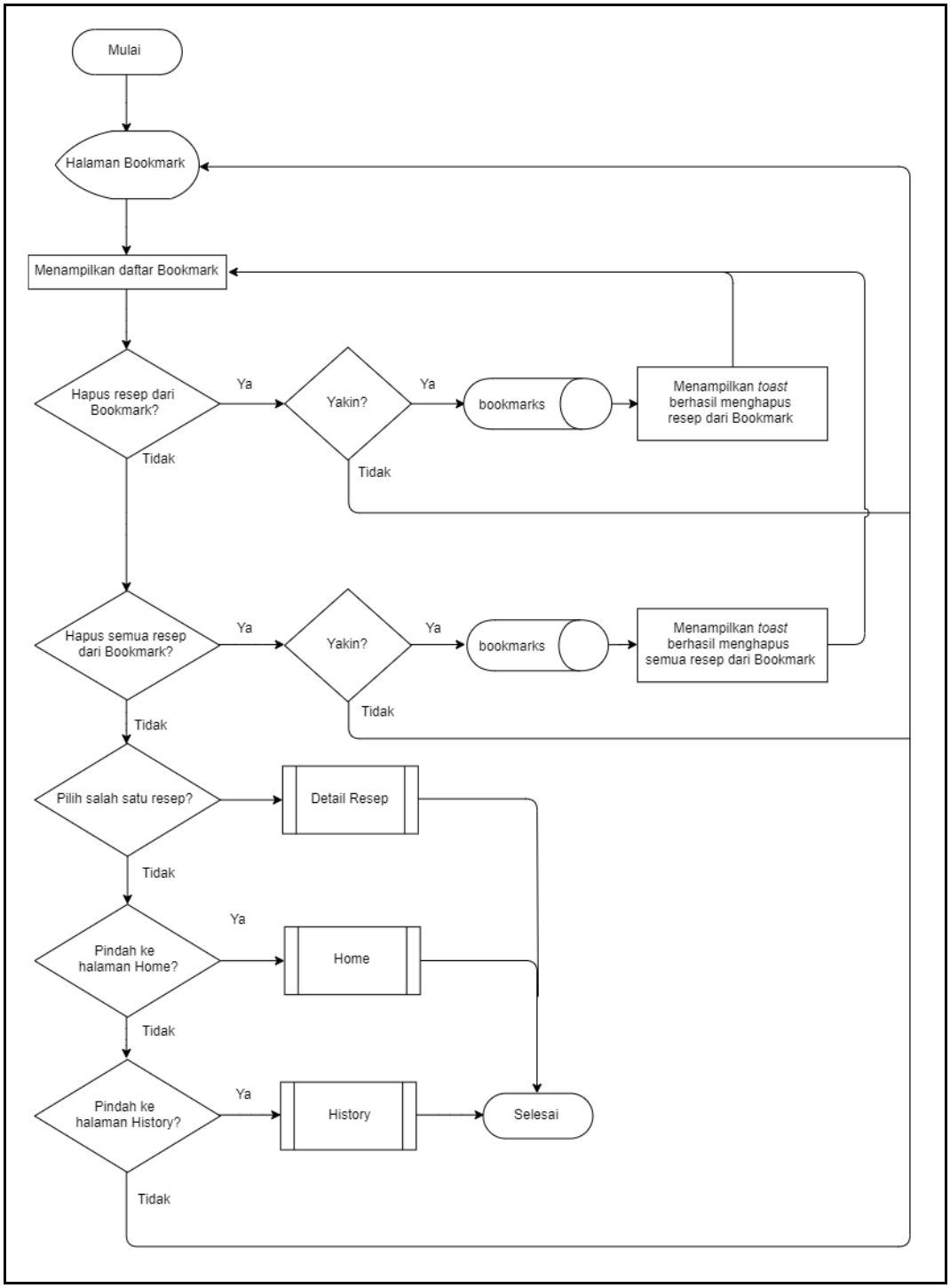
Gambar 3.19 merupakan Flowchart untuk halaman History. Ketika dari tab lain berpindah ke tab History, maka akan menampilkan tampilan History yang berisi daftar History atau resep-resep yang pernah dikunjungi sebelumnya. Di sini

terdapat menu untuk menghapus resep secara tunggal maupun semua resep. Daftar *history* disimpan dan diambil dari *local storage*. Sebelum menghapus resep, maka ada *alert* yang muncul untuk memvalidasi pengguna yakin untuk menghapus resep yang berkaitan serta memunculkan *toast* jika sudah berhasil menghapus.



Gambar 3.19 Flowchart History

Gambar 3.20 merupakan Flowchart untuk halaman Bookmark. Sama seperti halaman History, akan menampilkan tampilan Bookmark yang berisi daftar yang pernah di-*bookmark* sebelumnya. Di sini terdapat menu untuk menghapus resep secara tunggal maupun semua resep. Daftar *bookmark juga* disimpan dan diambil dari *local storage*. Sebelum menghapus resep, maka ada *alert* yang muncul untuk memvalidasi pengguna yakin untuk menghapus resep yang berkaitan dan menampilkan *toast*. Pada halaman ini juga bias mengarahkan pengguna langsung ke halaman Detail Resep ketika memilih salah satu daftar bookmark.

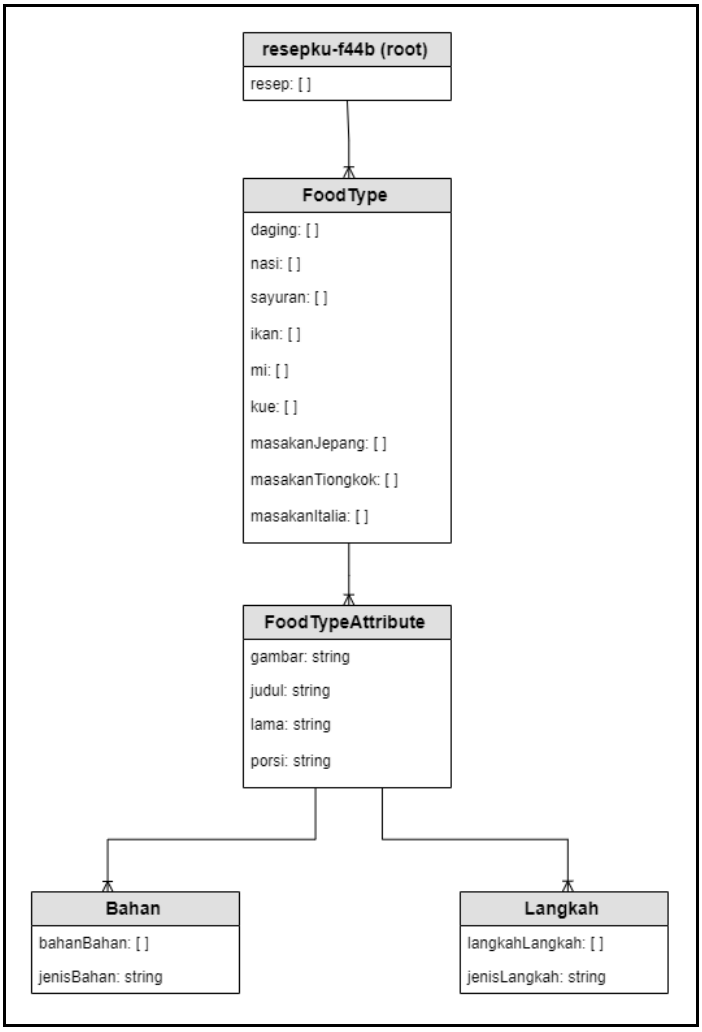


Gambar 3.20 Flowchart Bookmark

3.2.3 Conceptual Data Model

Untuk menggambarkan hubungan antara satu data dengan data yang lainnya maka dibutuhkan *data modelling*. Untuk menggambarkan model data ini digunakan

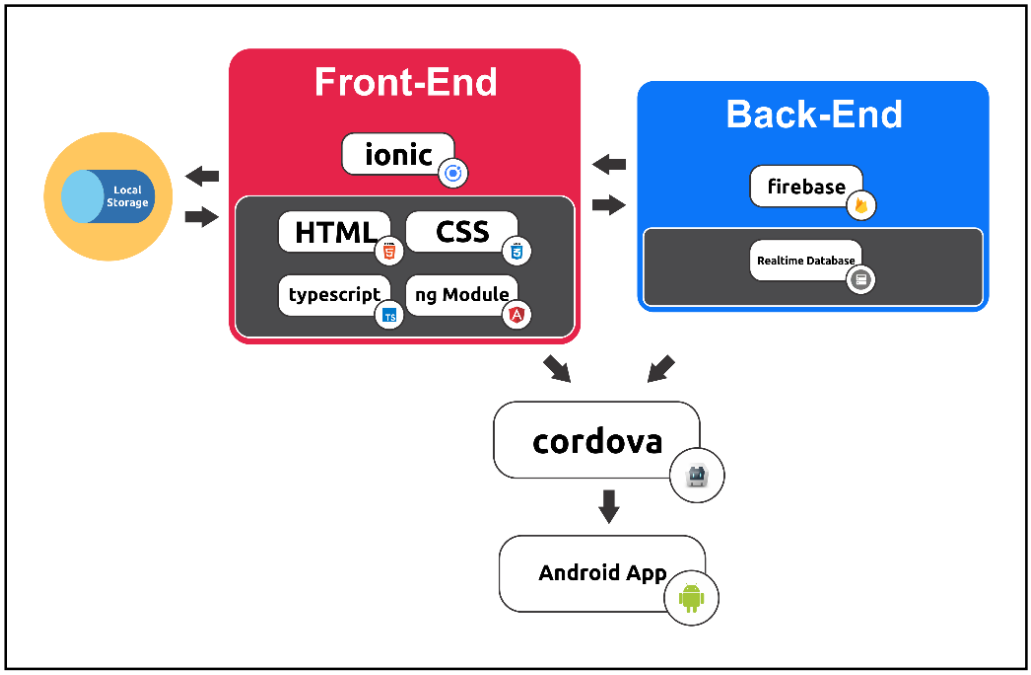
Conceptual Data Model yang ditunjukkan pada Gambar 3.21. Pada *data modelling* ini dimulai dari posisi paling atas yaitu root. Di dalam root terdapat entitas FoodType yang berisi jenis-jenis masakan, di mana masing-masing jenis tersebut berisi banyak resep. Untuk detail dari resep tersebut digambarkan dalam entitas FoodTypeAttribute. Terdapat dua bagian dalam FoodTypeAttribute yang memiliki lebih dari satu isi (array), yaitu Bahan dan Langkah. Semua tipe data dalam FoodTypeAttribute adalah string.



Gambar 3.21 Conceptual Data Model Risetta

3.2.4 Perancangan Aristektur Sistem

Arsitektur sistem digunakan untuk menggambarkan komponen-komponen apa saja yang digunakan beserta hubungannya antara satu dengan yang lainnya dalam sistem untuk mengembangkan aplikasi (Gambar 3.22).



Gambar 3.22 Arsitektur Sistem

Pada pengembangan aplikasi dibagi menjadi dua bagian, yaitu Front-End yang berhubungan langsung dengan tampilan User Interface (UI) dan Back-End yang menerima dan memberikan respon balik data yang diinginkan oleh Front-End. Pada bagian Front-End dengan bantuan framework Ionic, aplikasi mobile dapat didevelop dengan menggunakan teknologi web, dalam hal ini adalah HTML, CSS, typescript, dengan ng module dari Angular. Untuk bagian Back-End sudah disiapkan infrastruktur dari Firebase dengan teknologi Realtime Database. Pada aplikasi juga disiapkan local storage agar pengguna dapat mengakses dengan cepat data pribadi yang disimpan, seperti *bookmark* dan *history*. Kemudian ada teknologi

dari cordova yang berguna untuk melakukan proses *run* dan *build* agar aplikasi dapat dijalankan pada platform device lain, dalam hal penelitian ini adalah agar dapat dijalankan pada platform device Android.