

skripsi ziyad

by Muhammad Ziyad Al Maududi

Submission date: 29-Nov-2022 09:35AM (UTC+0700)

Submission ID: 1965786021

File name: Template-Skripsi-v3.0_1.docx (20.98M)

Word count: 23615

Character count: 151289

2 BAB 1 PENDAHULUAN

1.1 Latar Belakang

Memasak merupakan sebuah proses penciptaan suatu makanan dengan mengikuti langkah-langkah yang ada secara berurutan. Setiap makanan hasil masakan memiliki cita rasa yang berbeda-beda, tergantung bagaimana mengolahnya. Sangat tingginya tingkat antusiasme masyarakat terhadap kuliner sangat tinggi saat ini, karena selain memenuhi kebutuhan manusia akan gizi, namun telah menjadi sebuah objek rekreasi penghilang stres (Yasuma, 2017).

Berbagai media yang dapat digunakan dalam mencari resep masakan antara lain membaca buku mengenai resep masakan, menonton acara memasak di televisi, maupun mengikuti kursus memasak. Namun, sebagian orang menganggap cara-cara tersebut sangat merepotkan, sehingga mereka menggunakan *smartphone* untuk mengakses resep makanan tersebut via aplikasi (Karlina, Asian, & Mahmud, 2019). Selain itu, resep masakan yang hanya didapatkan secara turun-temurun terkadang menjadi terlupakan akibat padatnya aktivitas seseorang. Oleh karena itu, dibutuhkan resep masakan yang dapat diakses kapan pun saat dibutuhkan (Santoso, 2016). Di era globalisasi dengan semakin canggihnya kemampuan teknologi, berbagai resep masakan diposting setiap waktu oleh beberapa orang dari penjuru Indonesia melalui internet. Bahkan, kita dapat mengaksesnya tanpa harus menggunakan komputer, yaitu dengan menggunakan *smartphone/handphone*.

Beberapa aplikasi yang sudah ada terkait resep masakan antara lain Cookpad, Cookbook, Endeus, dan Resepedia (Mamduh, 2021). Namun, aplikasi-aplikasi tersebut menawarkan resep yang beragam. Aplikasi-aplikasi tersebut tidak spesifik terhadap sebuah kategori masakan, maupun jenis pengguna tertentu. Belum ada aplikasi yang menawarkan resep masakan yang dikhususkan secara spesifik untuk mahasiswa. Dalam skripsi ini dikembangkan aplikasi masakan yang ditujukan untuk mahasiswa, dimana resep masakan yang disajikan bukan masakan yang kompleks, melainkan praktis, sederhana, dan dengan bahan yang mudah didapat. Hal ini dibutuhkan karena beberapa permasalahan yang dialami mahasiswa terkait memasak antara lain mengenai bahan dan bumbu yang sulit didapat, cara memasak, tingkat kematangan, waktu memasak, dan takarannya, yang mana mewakili masakan yang terlalu kompleks.

Berdasarkan survei yang telah dilakukan terhadap 130 masyarakat Indonesia (Lampiran A.1) yang berusia 18 tahun ke atas, yang mana 95,6% responden pernah, terkadang, atau suka memasak, didapatkan hasil bahwa 95,3% responden belajar memasak dari resep maupun tutorial, baik dari internet, turun-temurun, diajarkan orang lain, maupun dari buku resep. 79,2% responden pernah menghadapi permasalahan saat memasak terkait bahan dan bumbu, resep, tingkat kematangan, cara memasak, waktu memasak, takaran, menentukan menu, dan hasil yang tidak sesuai ekspektasi. 97,7% responden

merasa akan terbantu dalam mengatasi permasalahan terkait memasak dengan adanya aplikasi yang memberikan resep masakan.

Survey lanjutan dilakukan terhadap 82 mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya (Lampiran A.2), yang mana 97,6% responden pernah, terkadang, atau suka memasak, didapatkan hasil bahwa 80,5% responden pernah belajar memasak dari resep maupun tutorial di internet. Tujuannya yaitu untuk memahami pengalaman mahasiswa saat mencari resep masakan, memasak, dan masalah apa yang dihadapi. 86,6% responden pernah menghadapi permasalahan saat memasak terkait bahan dan bumbu, resep, tingkat kematangan, cara memasak, waktu memasak, takaran, menentukan menu, dan hasil yang tidak sesuai ekspektasi. 96,3% responden merasa akan terbantu dalam mengatasi permasalahan terkait memasak dengan adanya aplikasi yang memberikan resep masakan.

Salah satu penelitian terkait penelitian ini pernah dilakukan Karlina, Asian, dan Mahmud (2019), dimana mereka membuat penelitian yang berjudul “Rancang Bangun Aplikasi Resep Masakan Menggunakan Metode XP Berbasis Android”. Pengembangan dalam penelitian tersebut menggunakan metode Extreme Programming, yang mana menghasilkan sebuah aplikasi yang membantu masyarakat mengetahui resep dan tata cara pembuatan masakan dengan mudah. Perbedaannya dengan penelitian ini yaitu pengembangan perangkat lunak menggunakan metode pengembangan Extreme Programming, sedangkan pada penelitian ini menggunakan metode Prototyping.

Metode SDLC Prototyping digunakan dalam pengembangan aplikasi ini karena lebih fleksibel dan dapat dilakukan penyesuaian dan perubahan kebutuhan perangkat lunak sesuai dengan kebutuhan pengguna dalam pengembangannya (Muchlison, 2021).

Alasan penggunaan aplikasi perangkat bergerak sebagai solusi dari permasalahan tersebut yaitu karena aplikasi perangkat bergerak lebih fleksibel, dan lebih mudah digunakan. Aplikasi perangkat bergerak juga dipilih pada lingkungan penelitian ini karena berdasarkan *survey* penggunaan TIK tahun 2017, 93,03% mahasiswa Diploma dan S1 telah menggunakan *smartphone*, dan 100% mahasiswa S2 dan S3 telah menggunakan *smartphone* (Kementerian Komunikasi dan Informatika, 2017). Pemilihan Android sebagai bentuk aplikasi perangkat bergerak dimaksudkan untuk menjangkau pengguna yang lebih banyak, dan didasari statistik mengenai jumlah pengguna sistem operasi Android di Indonesia yang mencapai 90,8% (Statcounter, 2022).

Sebagai solusi dari permasalahan tersebut, dalam skripsi ini dikembangkan sebuah aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa dengan metode SDLC Prototyping, sehingga diharapkan mampu membantu mahasiswa agar bisa memasak masakan sederhana dengan harga yang diinginkan.

1 **1.2 Rumusan Masalah**

Rumusan masalah dari penelitian pengembangan aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa berbasis Android adalah sebagai berikut:

1. Siapakah pengguna aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa berbasis Android?
2. Apa tujuan pengguna aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa berbasis Android?
3. Dapatkah aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa direalisasikan dalam bentuk aplikasi Android?
4. Dapatkah pengguna mencapai tujuannya dengan menggunakan aplikasi yang dikembangkan? **2**

1.3 Tujuan

Tujuan dari penelitian pengembangan aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa berbasis Android adalah sebagai berikut:

1. Mengetahui siapa saja pengguna aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa berbasis Android.
2. Mengetahui tujuan pengguna aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa berbasis Android.
3. Mengetahui apakah aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa dapat direalisasikan dalam bentuk aplikasi Android.
4. Mengetahui apakah pengguna dapat mencapai tujuannya dengan menggunakan aplikasi yang dikembangkan.

1.4 Manfaat

Manfaat dari penelitian pengembangan aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa berbasis Android adalah untuk memberikan resep dan panduan dalam memasak untuk mahasiswa, dalam hal ini masakan sederhana yang sesuai dengan mahasiswa dengan harga yang diinginkan.

1 **1.5 Batasan Masalah**

Batasan masalah dari penelitian pengembangan aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa berbasis Android adalah sebagai berikut:

1. Aplikasi ini dibuat untuk *platform* sistem operasi Android dengan API 21 sampai 32. Minimum API 21 dipakai karena *widget RecyclerView* mulai diperkenalkan pada API 21 (Android Developers, 2022).
2. Penggunaan aplikasi memerlukan koneksi internet.
3. Aplikasi ini dibuat menggunakan bahasa pemrograman Kotlin, berbasis Android, dan menggunakan Android Studio.

4. Mahasiswa yang dimaksud dalam penelitian ini memiliki studi kasus dalam lingkup mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya.

1

1.6 Sistematika Pembahasan

Gambaran mengenai penelitian ini dipaparkan dalam sistematika pembahasan yang terdiri dari beberapa bab yaitu sebagai berikut.

1. BAB 1 – PENDAHULUAN

Bab ini digunakan untuk menjelaskan latar belakang penelitian, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

2. BAB 2 - LANDASAN KEPUSTAKAAN

Bab ini berisi tentang kajian pustaka atau dasar teori yang digunakan dalam penyelesaian permasalahan di penelitian ini, teknologi pendukung, dan uraian hasil penelitian-penelitian terdahulu yang berhubungan dengan penelitian ini.

3. BAB 3 – METODOLOGI PENELITIAN

Bab ini berisi tentang cara sistematis dalam penyelesaian masalah, yang mana mencakup tipe penelitian, metode penelitian, subjek penelitian, lokasi penelitian, teknik pengumpulan data, peralatan pendukung yang digunakan, dan metode pengembangan.

4. BAB 4 - ANALISIS KEBUTUHAN

Bab ini berisi tentang deskripsi sistem, dan analisis kebutuhan perangkat lunak.

5. BAB 5 – PERANCANGAN

Bab ini berisi tentang perancangan sistem yang mengacu pada hasil analisis kebutuhan perangkat lunak.

6. BAB 6 – IMPLEMENTASI

Bab ini berisi tentang implementasi dari perancangan perangkat lunak.

7. BAB 7 – PENGUJIAN

Bab ini berisi tentang pengujian dari perangkat lunak.

8. BAB 8 – KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dan saran.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Dalam melakukan penelitian, langkah awal penyelesaian permasalahan yaitu dengan mencari tahu langkah-langkah yang diperlukan yang bersumber dari penelitian-penelitian terdahulu yang terkait dengan penelitian yang dilakukan (Muchlison, 2021). Berikut merupakan beberapa penelitian terdahulu yang terkait dengan penelitian ini.

Tabel 2.1 Penelitian terdahulu yang terkait

Penulis	Judul	Metode	Hasil
Lina Karlina Jelita Asian Muhamad Mahmud	Rancang Bangun Aplikasi Resep Masakan Menggunakan Metode XP Berbasis Android (2019)	Metode Extreme Programmin g	Aplikasi membantu masyarakat mengetahui resep dan tatacara pembuatan masakan dengan mudah
Ilham Dwi Muchlison	Pengembangan Aplikasi Perangkat Bergerak Sistem Informasi Event di Bidang Teknologi Informasi Berbasis Android (2021)	Metode Prototyping	Aplikasi membantu masyarakat untuk mendapatkan informasi terkait event di bidang teknologi informasi sesuai dengan topik yang diinginkan dan pengingat akan event yang akan di hadiri, dan mempermudah penyelenggara event menyebarluaskan informasi terkait event yang diadakan dengan menjangkau peserta yang sesuai dengan topik dibidangnya.
Pristi Sukmasetya	Penggunaan	Metode	Penelitian ini

Agus Setiawan Endah Ratna Arumi	10 <i>Usability Testing Sebagai Alat Evaluasi Website KRS Online pada Perguruan Tinggi (2020)</i>	<i>Usability Testing</i>	mengevaluasi website KRS online yang terdapat pada perguruan tinggi dengan <i>in depth interview</i> dan <i>usability testing</i> untuk mengetahui tingkat kebergunaan dari sistem tersebut.
12 Dhiki Sekti Wibawa Yusi Tyroni Mursityo Retno Indah Rokhmawati	1 <i>Evaluasi Usability dan Perbaikan Antarmuka Pengguna Aplikasi Mobile Malang Menyapa Menggunakan Metode Usability Testing (2019)</i>	<i>Usability Testing</i>	Penelitian ini melakukan evaluasi dan perbaikan antarmuka pengguna pada aplikasi mobile Malang Menyapa menggunakan metode <i>usability testing</i> untuk menemukan permasalahan <i>usability</i> yang dialami pengguna dan melakukan evaluasi untuk menyusun perbaikan antarmuka aplikasi tersebut.

14
Pada penelitian berjudul "Rancang Bangun Aplikasi Resep Masakan Menggunakan Metode XP Berbasis Android", dihasilkan sebuah aplikasi resep dengan menggunakan metode Extreme Programming, dimana perbedaannya dengan penelitian ini yaitu penelitian ini mengembangkan sebuah aplikasi resep menggunakan metode Prototyping. Sedangkan persamaannya yaitu keduanya menghasilkan sebuah aplikasi resep.

1
Pada penelitian berjudul "Pengembangan Aplikasi Perangkat Bergerak Sistem Informasi Event di Bidang Teknologi Informasi Berbasis Android", dihasilkan sebuah aplikasi sistem informasi *event* dengan menggunakan metode Prototyping, dimana perbedaannya dengan penelitian ini yaitu penelitian ini mengembangkan sebuah aplikasi resep. Sedangkan persamaannya yaitu keduanya menggunakan metode Prototyping.

Pada penelitian berjudul "Penggunaan *Usability Testing* Sebagai Alat Evaluasi Website KRS *Online* pada Perguruan Tinggi", dilakukan evaluasi terhadap website KRS *online* perguruan tinggi dengan menggunakan *usability testing*, dimana perbedaannya dengan penelitian ini yaitu penelitian ini mengembangkan sebuah aplikasi resep. Sedangkan persamaannya yaitu keduanya menggunakan metode *usability testing* pada tahap pengujian.

Pada penelitian berjudul "Evaluasi *Usability* dan Perbaikan Antarmuka Pengguna Aplikasi *Mobile Malang Menyapa* Menggunakan Metode *Usability Testing*", dilakukan evaluasi dan perbaikan antarmuka pengguna terhadap aplikasi mobile Malang Menyapa menggunakan metode *usability testing*, dimana perbedaannya dengan penelitian ini yaitu penelitian ini mengembangkan sebuah aplikasi resep. Sedangkan persamaannya yaitu keduanya menggunakan metode *usability testing* pada tahap pengujian.

2.2 Resep Masakan

Resep masakan adalah sebuah takaran yang digunakan untuk mengolah bahan makanan yang keakuratannya telah teruji. Takaran yang dimaksud dapat berupa bumbu, bahan, maupun cara pengolahan bahan makanan (Arifien, 2022). Berbagai media yang dapat digunakan dalam mencari resep masakan antara lain membaca buku mengenai resep masakan, menonton acara memasak di televisi, maupun mengikuti kursus memasak. Namun, sebagian orang menganggap cara-cara tersebut sangat merepotkan, sehingga mereka menggunakan *smartphone* untuk mengakses resep makanan tersebut via aplikasi (Karlina, Asian, & Mahmud, 2019). Selain itu, resep masakan yang hanya didapatkan secara turun-temurun terkadang menjadi terlupakan akibat padatnya aktivitas seseorang. Oleh karena itu, dibutuhkan resep masakan yang dapat diakses kapan pun saat dibutuhkan (Santoso, 2016).

2.2.1 Resep Masakan untuk Mahasiswa

Makanan yang terbaik yaitu yang dimasak sendiri karena cara memasak bahan-bahan, kandungan gizi, dan tingkat higienisnya dapat dipastikan sendiri. Namun, mahasiswa seringkali tidak akan memiliki waktu yang cukup dalam memasak makanannya sendiri, kecuali menu yang dimasak mudah dan sederhana (Ramadana, 2015). Meskipun memasak sendiri lebih hemat, terkadang mahasiswa bingung menentukan masakan yang ingin dimasak, selain dari kemalasannya untuk memasak. Namun, memasak makanan sendiri lebih praktis, dan lebih aman karena dapat memilih bahan makanan yang ingin digunakan dalam memasak. Masakan yang sering dimasak mahasiswa adalah masakan simpel ala mahasiswa, antara lain telor ceplok, sayur tauge, dan tahu tumis kecap. Pemilihan menu tersebut didasari pada kemudahan dalam memperoleh bahan yang dibutuhkan karena dapat diperoleh di penjual sayur maupun pasar terdekat (Ervina, 2020).

Salah satu buku terkait resep masakan untuk mahasiswa pernah diterbitkan dengan judul "My Little College Cookbook: Easy College Recipes for Students!"

yang berisi resep-resep unik untuk anak kuliah (BookSumo Press, 2021). Pada buku resep tersebut, disajikan berbagai resep masakan yang mudah, berfokus pada masakan mahasiswa, sederhana, namun sangat unik. Selain itu, meskipun sederhana, rasa dari resep tersebut cukup menakjubkan dan hanya membutuhkan sedikit usaha. Selain itu, ada juga buku resep terkait masakan praktis berjudul "Menu Praktis untuk Sebulan" yang berisi menu praktis dengan waktu memasak yang relatif tidak lama (Kinanthi, 2013). Pada buku resep tersebut, terdapat beberapa masakan praktis sehari-hari.

2.3 SDLC

²⁴ Pengembangan perangkat lunak adalah sebuah proses membuat sebuah perangkat lunak baru untuk menggantikan yang lama secara keseluruhan, atau memperbaiki perangkat lunak yang sudah ada. Dibutuhkan sebuah metodologi khusus agar solusi dan pengembangan perangkat lunak lebih cepat, tepat, dan mudah dikembangkan dan dipelihara. Metodologi pengembangan perangkat lunak merupakan suatu proses pengorganisasian konvensi notasi, dan kumpulan metode yang telah didefinisikan dalam mengembangkan suatu perangkat lunak yang bertujuan menghasilkan perangkat lunak yang berkualitas (Academia, 2022).

Software Development Life Cycle atau dapat disingkat SDLC adalah sebuah metode pengembangan perangkat lunak yang berisi analisis kebutuhan perangkat lunak, desain, pembuatan kode program, pengujian, dan pemeliharaan. Dalam pengembangan perangkat lunak, dibutuhkan SDLC untuk mengurangi terjadinya krisis perangkat lunak. Hal itu dikarenakan metode SDLC memiliki tahapan yang sangat detail, mulai dari analisis, hingga pemeliharaan. SDLC juga dapat memudahkan programmer mengembangkan sistem, mengurangi kesalahan, dan mempercepat pengerjaan (Aisah, et al., 2021).

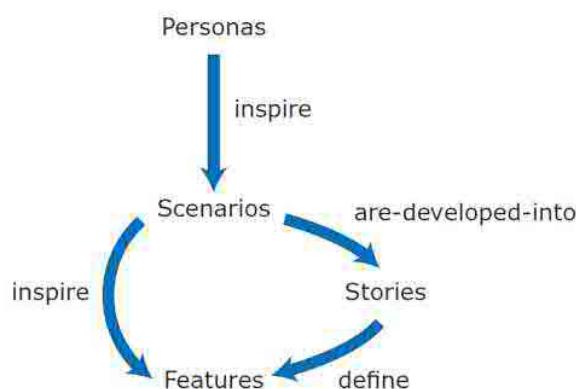
2.4 Prototyping

Prototyping adalah sebuah proses pengembangan dari versi awal produk untuk menguji ide, mengecek produk yang dibuat, dan meyakinkan pengembang dan perusahaan bahwa produk tersebut benar-benar potensial. Prototyping juga membantu pengembang memahami struktur, dan bagaimana mengatur versi akhir dari produk. Sebuah *prototype* dapat membantu memahami komponen dan layanan perangkat lunak yang diperlukan, dan menguji teknologinya. Terkadang teknologi yang direncanakan untuk dipakai tidak memadai, dan harus melakukan revisi ide dalam mengimplementasikan perangkat lunak (Sommerville, 2021).

Membuat *prototype* merupakan hal pertama yang harus dilakukan saat mengembangkan sebuah produk perangkat lunak. Tujuannya adalah agar tersedia sebuah versi bekerja dari perangkat lunak yang dapat mendemonstrasikan fitur-fitur kunci yang ada. Singkatnya siklus pengembangan merupakan hal kritis dalam Prototyping, dimana kegiatan demonstrasi dan menjalankan sistem ditargetkan selesai dalam 4 hingga 6 minggu, dengan

kompensasi mengabaikan masalah seperti keandalan dan kinerja, serta menggunakan antarmuka pengguna yang belum sempurna (Sommerville, 2021).

Prototyping terdiri dari 2 tahapan proses, yaitu demonstrasi kelayakan, dan demonstrasi pelanggan. Pada tahap demonstrasi kelayakan, dibuat sebuah sistem yang dapat dijalankan untuk mendemonstrasikan ide baru dari produk. Tujuannya untuk melihat apakah ide tersebut benar benar bekerja, dan menunjukkan bahwa fitur produk tersebut lebih baik daripada produk pesaing. Pada tahap demonstrasi pelanggan, diambil *prototype* yang telah dibuat untuk mendemonstrasikan kelayakan, dan menambahkannya dengan ide baru untuk fitur pelanggan secara spesifik, beserta cara mewujudkannya. (Sommerville, 2021).



Gambar 2.1 Tahapan dari *persona* hingga menjadi *feature*

Sumber: Sommerville (2021)

Sebelum membuat *prototype* pelanggan, dibutuhkan beberapa studi terhadap pengguna dan ide yang jelas mengenai pengguna yang potensial, dan skenario yang digunakan. Untuk itu, dibutuhkan *persona* dan *scenario*, yang mana nantinya akan menghasilkan *user story* untuk dipakai dalam penentuan *feature*. Dalam mengembangkan *prototype*, diharuskan agar selalu menggunakan teknologi yang dikuasai, sehingga tidak harus menghabiskan waktu dalam mempelajari bahasa atau *framework* baru. Kekokohan arsitektur perangkat lunak, fitur keamanan, dan keandalan perangkat lunak, bukanlah sebuah keharusan dalam Prototyping (Sommerville, 2021).

Evaluasi terhadap *prototype* dapat dilakukan menggunakan studi pengguna kualitatif. Studi pengguna kualitatif biasanya bertujuan mencari komponen yang tak berfungsi pada desain, memperbaikinya, dan melanjutkan dengan desain baru yang lebih baik. Studi pengguna kualitatif seringkali hanya bertujuan menyempurnakan iterasi desain dan mengidentifikasi permasalahan terkait kegunaan, bukan memprediksi jumlah pengguna yang akan menyelesaikan *task*, dan jumlah pengguna yang mengalami permasalahan. Pengujian pada studi pengguna kualitatif direkomendasikan dengan jumlah pengguna 5 orang. Jumlah

tersebut berdasar pada asumsi bahwa tujuan dari pengujian ini adalah mengidentifikasi masalah pada desain, masalah apapun yang dialami pengguna adalah masalah valid dan layak diperbaiki, dan kemungkinan pengguna mengalami masalah adalah 31%. Berdasarkan asumsi tersebut, Jakob Nielsen dan Tom Landauer membuat sebuah model matematika dan melakukan pengujian ini dengan 5 partisipan, dan menemukan bahwa 85% masalah pada antarmuka akan teridentifikasi. Tak ada gunanya mencoba mencari semua masalah hanya dalam 1 pengujian karena hanya akan menghabiskan banyak uang dan waktu, dan pasti akan muncul masalah lain saat membuat desain ulang. Sedangkan, untuk studi kuantitatif, jumlah yang direkomendasikan biasanya lebih dari 30 orang (Budiu, 2021).

2.5 Persona

Dalam mengembangkan sebuah produk perangkat lunak, diharuskan untuk mengerti pengguna potensial dalam mendesain fitur yang dirasa berguna, dan mendesain sebuah antarmuka yang cocok untuk mereka. Namun, terkadang dalam pengembangan dapat terjadi inkonsistensi pandangan dalam implementasi perangkat lunak. Oleh karena itu, dibutuhkan sebuah *shared vision* atau bayangan bersama mengenai pandangan, bakat, dan motivasi dari pengguna dalam menggunakan perangkat lunak yang akan dikembangkan. Dalam hal ini, dapat digunakan *persona* untuk merepresentasikan *shared vision* tersebut (Sommerville, 2021).

Persona adalah sebuah ‘pengguna khayalan’, potret karakter dari tipe pengguna yang mungkin akan menggunakan produk tersebut. Sebagai contoh, pada sebuah aplikasi pemesanan dokter gigi, akan terdapat 3 macam *persona* yaitu dokter gigi, resepionis, dan pasien. *Persona* untuk tipe pengguna yang berbeda akan membantu membayangkan apa saja yang mungkin pengguna inginkan dari perangkat lunak, bagaimana mereka menggunakan, dan kesulitan yang mungkin akan dialami dalam memahami dan menggunakan fitur (Sommerville, 2021).



Gambar 2.2 Deskripsi sebuah *persona*

Sumber: Sommerville (2021)

Beberapa rekomendasi aspek dari sebuah deskripsi persona antara lain personalisasi, relevansi, edukasi, dan pekerjaan. Aspek personalisasi merupakan sebuah informasi dan keadaan personal dari sebuah *persona*. Aspek pekerjaan merupakan sebuah informasi detail terkait pekerjaan dari sebuah *persona*. Aspek Edukasi merupakan sebuah informasi terkait edukasi, kemampuan teknis, dan pengalaman dari sebuah *persona*. Aspek relevansi merupakan sebuah informasi terkait detail dari kemungkinan penyebab pengguna tertarik menggunakan produk, dan apa yang mereka ingin lakukan dengan produk tersebut (Sommerville, 2021).

Dalam membuat sebuah *persona*, haruslah didasarkan pada pemahaman terkait pengguna potensial dari produk, beserta pekerjaan, latar belakang, dan aspirasi mereka. Dibutuhkan studi, dan potensial untuk mengetahui apa yang mereka butuhkan, dan bagaimana mereka menggunakan produk. Dari data tersebut, dapat dibuat sebuah abstraksi dari informasi penting terkait berbagai jenis pengguna produk, dan menggunakannya sebagai dasar pembuatan *persona*. *Persona* yang dibuat harus diperiksa ulang terhadap data pengguna untuk memastikan bahwa *persona* tersebut mencerminkan tipikal pengguna produk (Sommerville, 2021). Sebelum nantinya membuat *user story*, diharuskan untuk memahami target pengguna dengan menggunakan *persona*. Hal itu karena pada *user story* menceritakan kisah pengguna dalam menggunakan produk, sehingga dapat diketahui siapa pengguna, masalah yang ingin dipecahkan, dan daftar keinginan dari deskripsi fungsi produk yang relevan. *Persona* menawarkan cara yang bagus dalam menggambarkan pengguna dengan kebutuhannya (Romanpichler, 2014).

Emma, a history teacher

Emma, age 41, is a history teacher in a secondary school (high school) in Edinburgh. She teaches students from ages 12 to 18. She was born in Cardiff in Wales, where both her father and her mother were teachers. After completing a degree in history from Newcastle University, she moved to Edinburgh to be with her partner and trained as a teacher. She has two children, aged 6 and 8, who both attend the local primary school. She likes to get home as early as she can to see her children, so often does lesson preparation, administration, and marking from home.

Emma uses social media and the usual productivity applications to prepare her lessons, but is not particularly interested in digital technologies. She hates the virtual learning environment that is currently used in her school and avoids using it if she can. She believes that face-to-face teaching is most effective. She might use the iLearn system for administration and access to historical films and documents. However, she is not interested in a blended digital/face-to-face approach to teaching.

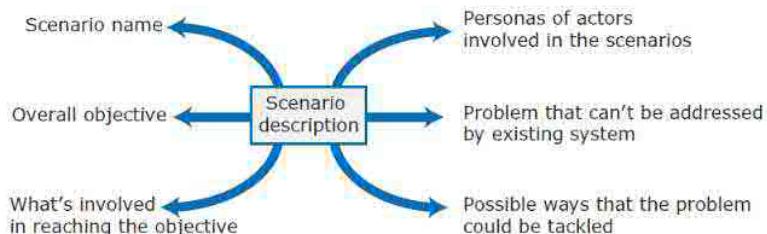
Gambar 2.3 Contoh *persona* dalam bentuk deskripsi naratif

Sumber: Sommerville (2021)

Pada gambar 2.3 terdapat sebuah contoh *persona* dalam bentuk deskripsi naratif, dimana *persona* tersebut mencakup aspek personalisasi, relevansi, edukasi, dan pekerjaan. Aspek personalisasi diwakili oleh deskripsi personal Emma. Aspek relevansi diwakili oleh ketertarikannya dalam menggunakan *iLearn system* untuk administrasi dan akses terhadap film dan dokumen sejarah. Aspek edukasi diwakili dengan riwayat pendidikan Emma. Aspek pekerjaan diwakili oleh deskripsi pekerjaan Emma.

2.6 Scenario

Scenario adalah sebuah narasi mengenai situasi yang mana pengguna menggunakan *feature* atau fitur dari produk untuk melakukan hal yang mereka inginkan. Secara singkat, *scenario* haruslah memaparkan masalah pengguna, dan menunjukkan cara yang dibayangkan dalam penyelesaian masalah tersebut. Menyertakan segala hal bukanlah merupakan keharusan dalam membuat *scenario*, karena hal tersebut tidak menjelaskan spesifikasi sistem (Sommerville, 2021).



Gambar 2.4 Deskripsi sebuah *scenario*

Sumber: Sommerville (2021)

Scenario mencakup nama skenario, tujuan keseluruhan, hal yang terlibat dalam mencapai tujuan, persona dari aktor yang terlibat, permasalahan yang

tidak bisa terselesaikan oleh sistem yang sudah ada, dan cara yang memungkinkan dalam menyelesaikan permasalahan tersebut.

Scenario digunakan dalam tahap desain persyaratan dan fitur sistem, pengujian, dan desain antarmuka pengguna. *Scenario* juga memfasilitasi komunikasi, membantu pengembang mendapatkan pemahaman bersama terkait sistem yang diciptakan, dan merangsang kreativitas desain. Dikatakan efektif dalam komunikasi karena dapat dimengerti, dan diakses oleh pengguna dan pemberi dana. Terdapat dua macam *scenario* yang biasa dipakai, yaitu *scenario naratif*, dan *scenario struktural*. *Scenario* naratif lebih direkomendasikan karena lebih mudah untuk dimengerti oleh pengguna sistem. Sebuah *scenario* naratif biasanya dapat terdiri dari dua hingga tiga paragraf teks. Namun, terkadang diperlukan deskripsi yang lebih panjang jika perangkat lunak akan digunakan pada proses yang telah ada, dan digunakan bersama perangkat lunak lain. Pada kasus ini, mungkin diperlukan juga deskripsi interaksi dengan proses lain, dan perangkat lunak sistem (Sommerville, 2021).

Fishing in Ullapool

Jack is a primary school teacher in Ullapool, teaching P6 pupils. He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development, and economic impact of fishing.

As part of this, students are asked to gather and share reminiscences from relatives, use newspaper archives, and collect old photographs related to fishing and fishing communities in the area. Pupils use an iLearn wiki to gather together fishing stories and SCARAN (a history archive site) to access newspaper archives and photographs. However, Jack also needs a photo-sharing site as he wants students to take and comment on each others' photos and to upload scans of old photographs that they may have in their families. He needs to be able to moderate posts with photos before they are shared, because pre-teen children can't understand copyright and privacy issues.

Jack sends an email to a primary school teachers' group to see if anyone can recommend an appropriate system. Two teachers reply and both suggest that he use KidsTakePics, a photo-sharing site that allows teachers to check and moderate content. As KidsTakePics is not integrated with the iLearn authentication service, he sets up a teacher and a class account with KidsTakePics.

He uses the iLearn setup service to add KidsTakePics to the services seen by the students in his class so that, when they log in, they can immediately use the system to upload photos from their phones and class computers.

Gambar 2.5 Contoh *scenario*

Sumber: Sommerville (2021)

Pada gambar 2.5 terdapat sebuah contoh *scenario*, dimana *scenario* tersebut mencakup nama *scenario*, tujuan keseluruhan, hal yang terlibat dalam mencapai tujuan, persona dari aktor yang terlibat, permasalahan yang tidak bisa terselesaikan oleh sistem yang sudah ada, dan cara yang memungkinkan dalam menyelesaikan permasalahan tersebut. Dalam hal ini, *scenario* tersebut menceritakan tentang memancing di Ullapool, dimana Jack sebagai persona yang terlibat, merupakan guru di Ullapool. Jack bertujuan memfokuskan projek kelas pada area industri pemancingan. Hal-hal yang terlibat antara lain pengumpulan informasi oleh siswa terkait pemancingan dan komunitas pemancingan, dan pengumpulan cerita pemancingan untuk mengakses arsip koran dan fotografi. Permasalahan yang belum terselesaikan yaitu ia membutuhkan situs pembagian

foto, berkomentar, dan mengunggah foto lama yang mungkin dimiliki oleh keluarganya. Di lain sisi, melakukan moderasi postingan foto tersebut karena alasan hak cipta dan privasi. Cara yang memungkinkan untuk menyelesaikan permasalahan tersebut yaitu dengan menggunakan iLearn setup service untuk menambahkan KidsTakePics, sehingga layanan tersebut dapat dilihat oleh murid-murid di kelas dan saat mereka *login*, mereka dapat menggunakan sistem secara langsung untuk mengunggah foto dari *handphone* mereka dan komputer kelas.

2.7 User Story

User Story adalah sebuah deskripsi mengenai situasi dimana pengguna mencoba melakukan sesuatu dengan sistem perangkat lunak. *Scenario* merupakan *user story* tingkat tinggi yang dipakai oleh sistem. *User story* haruslah menjelaskan urutan interaksi dengan sistem, namun tidak termasuk detail dari interaksi tersebut. Selain itu, *user story* juga dapat diartikan sebagai narasi yang lebih halus, lebih rinci, dan terstruktur, terkait cara sebuah hal yang diinginkan pengguna dari sistem perangkat lunak (Sommerville, 2021).

Salah satu penggunaan penting dari *user story* adalah untuk perencanaan, dimana *user story* harus berfokus pada kejelasan fitur sistem yang ditentukan, atau aspek dari fitur yang dapat diimplementasikan. Saat menentukan *user story* dari sebuah *scenario*, lebih banyak informasi akan disediakan untuk pengembang untuk membantu mereka mendesain fitur dari produk. *Scenario* dan *user story* sangat berguna dalam memilih dan mendesain fitur pada sistem, namun hanya sebagai ‘alat untuk berpikir’, dan bukan sebagai spesifikasi sistem. *Scenario* dan *user story* hanya dipakai untuk menstimulasikan cara berpikir, yang mana tidak harus selesai ataupun konsisten, dan tidak ada aturan tetap terkait berapa banyak *scenario* dan *user story* yang dibutuhkan.

As a <role>, I <want / need> to <do something>

Gambar 2.6 Format standar kalimat *user story*

Sumber: Sommerville (2021)

Pada gambar 2.6 terdapat sebuah format standar yang dapat digunakan sebagai kalimat *user story*. Format tersebut yaitu “Sebagai seorang <peran>, saya <ingin/butuh> (untuk) <lakukan sesuatu>”.

As an author I need a way to organize the book that I'm writing into chapters and sections.

Gambar 2.7 Contoh kalimat *user story* dengan format standar

Sumber: Sommerville (2021)

Pada gambar 2.7 terdapat sebuah contoh kalimat *user story* dengan menggunakan format standar yaitu “Sebagai seorang penulis, saya membutuhkan sebuah cara untuk mengatur buku yang saya tulis menjadi bab-bab dan bagian-bagian”.

As a <role> I <want / need> to <do something> so that <reason>

Gambar 2.8 Format varian standar kalimat *user story* dengan justifikasi

Sumber: Sommerville (2021)

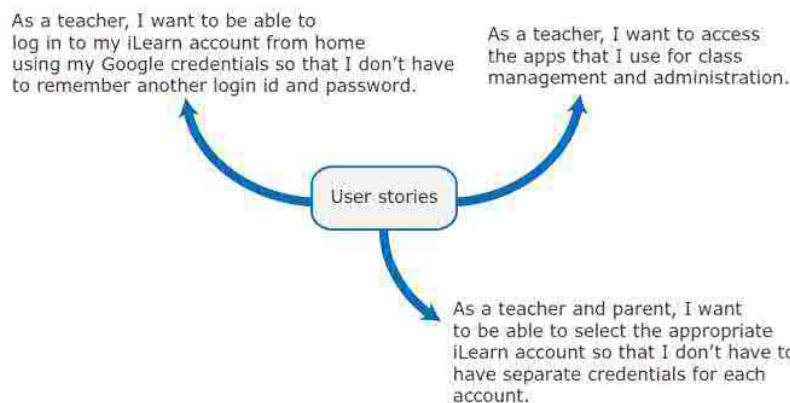
Pada gambar 2.8 terdapat sebuah varian lain dari format standar yang dapat digunakan sebagai kalimat *user story* disertai dengan justifikasi. Format tersebut yaitu "Sebagai seorang <peran>, saya <ingin/butuh> (untuk) <lakukan sesuatu> sehingga <alasan>".

As a teacher, I need to be able to report who is attending a class trip so that the school maintains the required health and safety records.

Gambar 2.9 Contoh kalimat *user story* dengan format varian standar

Sumber: Sommerville (2021)

Pada gambar 2.9 terdapat sebuah contoh kalimat *user story* dengan menggunakan format varian standar yaitu "Sebagai seorang guru, saya ingin dapat melaporkan siapa saja yang menghadiri perjalanan kelas sehingga sekolah mengurus catatan kesehatan dan keamanan yang dibutuhkan".

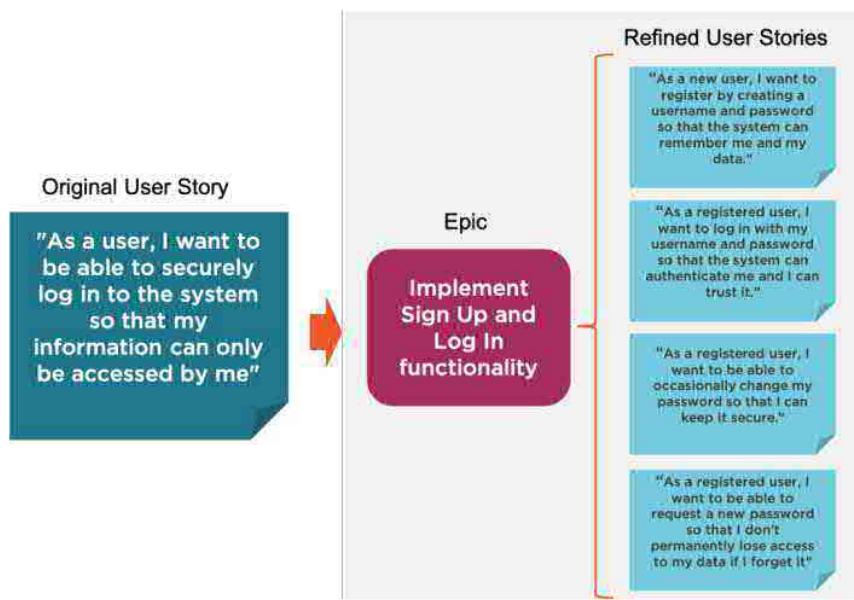


Gambar 2.10 Contoh *user story* dari sebuah scenario

Sumber: Sommerville (2021)

Pada gambar 2.10 terdapat sebuah contoh beberapa kalimat *user story* dengan menggunakan format standar dan format varian standar.

Selain *user story* standar dan variannya, dapat pula digunakan *user story* yang telah disempurnakan atau bisa juga disebut *refined user story*. *User story* ini lebih kecil, lebih tepat, dan disempurnakan cakupannya dan lebih mungkin untuk sesuai untuk masing-masing *user story*. Contohnya pada sebuah *user story* autentikasi akun dapat dipecah menjadi *user story* *login* dan *register* (Pluralsight, 2020).

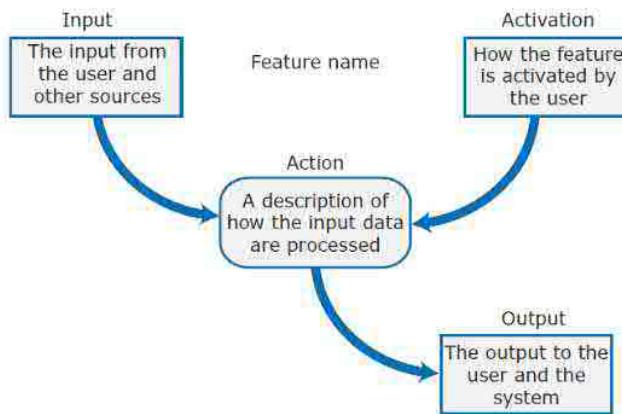


Gambar 2.11 Contoh refined user story

Pada gambar 2.11 terdapat sebuah contoh sebuah *user story* autentikasi akun yang dipecah menjadi *user story login* dan *register*.

2.8 Feature

Feature atau dapat juga disebut fitur adalah sebuah cara yang memungkinkan pengguna mengakses dan menggunakan fungsionalitas produk, sehingga daftar fitur mendefinisikan fungsionalitas sistem secara keseluruhan. Mengidentifikasi fitur dari sebuah produk haruslah independen, koheren, dan relevan. Independen berarti sebuah fitur seharusnya tidak tergantung kepada bagaimana fitur sistem lain diimplementasikan, dan tidak terpengaruh oleh urutan aktivasi fitur lain. Koheren berarti fitur haruslah terhubung dengan satu item fungsionalitas, tidak boleh melakukan lebih dari satu hal, dan haruslah tidak memiliki efek samping. Relevan berarti fitur sistem haruslah mencerminkan cara yang biasanya pengguna melakukan beberapa tugas, dan tidak boleh menawarkan fungsionalitas tidak jelas yang jarang diperlukan (Sommerville, 2021).



Gambar 2.12 Deskripsi sebuah fitur

Sumber: Sommerville (2021)

Pada gambar 2.12 terdapat deskripsi dari sebuah fitur. Fitur mencakup nama fitur, *input*, *activation*, *action*, dan *output*. *Input* merupakan masukan dari pengguna dan sumber lain. *Activation* merupakan bagaimana sebuah fitur teraktivasi oleh user. *Action* merupakan deskripsi bagaimana data dari *input* diproses. *Output* merupakan hasil atau keluaran dari fitur ke pengguna dan sistem.

<*action*> the <*result*> <by|for|of|to> <*object*>

Gambar 2.13 Format kalimat deskripsi sebuah fitur

Sumber: Sommerville (2021)

Pada gambar 2.13 terdapat sebuah format yang dapat digunakan sebagai kalimat deskripsi sebuah fitur. Format tersebut yaitu "<*aksi*> (sebuah) <*hasil*> <oleh|untuk|dari|kepada> <*objek*>".

Jack is a primary school teacher in Ullapool, teaching P6 pupils. He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development, and economic impact of fishing.

As part of this, students are asked to gather and share reminiscences from relatives, use newspaper archives, and collect old photographs related to fishing and fishing communities in the area. *Students use an iLearn wiki to gather together fishing stories and SCRAN (a history archive) to access newspaper archives and photographs.* However, Jack also needs a photo-sharing site as he wants *pupils to take and comment on each others' photos and to upload scans of old photographs* that they may have in their families. He needs to be able to moderate posts with photos before they are shared, because pre-teen children can't understand copyright and privacy issues.

Jack sends an email to a primary school teachers' group to see if anyone can recommend an appropriate system. Two teachers reply and both suggest that he use KidsTakePics, a photo-sharing site that allows teachers to check and moderate content. As KidsTakePics is not integrated with the iLearn authentication service, he sets up a teacher and a class account with KidsTakePics.

He uses the the iLearn setup service to add KidsTakePics to the services seen by the students in his class so that when they log in, they can immediately use the system to upload photos from their phones and class computers.

Gambar 2.14 Melakukan *highlight* frasa terhadap scenario

Sumber: Sommerville (2021)

Pada gambar 2.14 dapat dilihat bahwa fitur bisa didapatkan dari melakukan *highlight* terhadap deskripsi naratif pada *scenario* untuk menemukan fitur sistem. Caranya yaitu membaca *scenario*, mencari tindakan pengguna yang biasanya dilambangkan dengan kata kerja aktif seperti 'menggunakan', 'memilih', 'mengirim', dan 'memperbaharui', dan sorot frasa dimana kalimat tersebut disebutkan. Langkah selanjutnya yaitu menentukan dan mengidentifikasi fitur produk yang dapat mendukung tindakan ini, beserta cara penerapannya. Dalam mengidentifikasi fitur, dapat dilakukan dengan cara menggeneralisir set dari fitur produk. Jika *user story* dikembangkan untuk menyempurnakan *scenario*, maka fitur produk atau karakteristik fitur dapat lebih mudah didapatkan dari *scenario* tersebut (Sommerville, 2021).

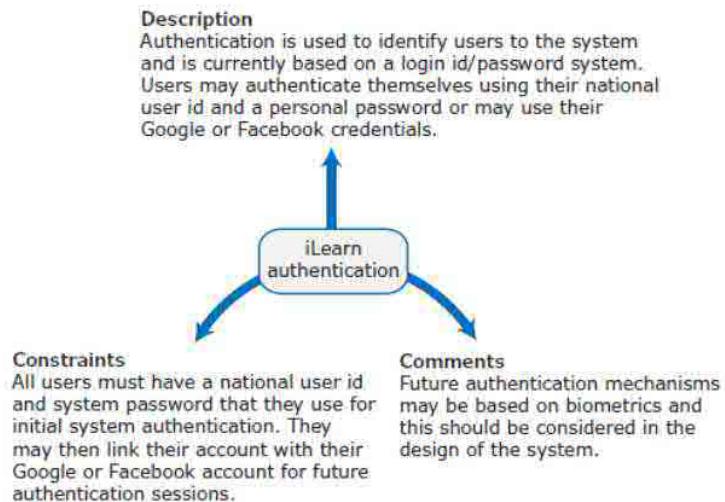
As a teacher and a parent, I want to be able to select the appropriate iLearn account so that I don't have to have separate credentials for each account.

Gambar 2.15 Contoh penentuan fitur berdasarkan *user story*

Sumber: Sommerville (2021)

Pada gambar 2.15 terdapat sebuah contoh sebuah *user story* yang nantinya dapat digunakan dalam menentukan sebuah fitur.

Setelah melakukan identifikasi fitur, seharusnya telah didapatkan daftar fitur yang ingin didesain dan diimplementasikan pada produk. Tidak perlu menambahkan detail pada daftar fitur, karena detail seharusnya ditambahkan saat mengimplementasikan fitur. Fitur pada daftar dapat digambarkan menggunakan Model *input/action/output*, atau dapat juga menggunakan deskripsi naratif.



Gambar 2.16 Contoh template fitur

Sumber: Sommerville (2021)

Pada gambar 2.16 terdapat salah satu contoh template fitur yang mana berisi description, constraints, dan comments. Description merupakan deskripsi dan kegunaan dari fitur tersebut. Constraints merupakan batasan yang terdapat pada sebuah fitur. Comments merupakan sebuah saran atau komentar terhadap fitur tersebut.

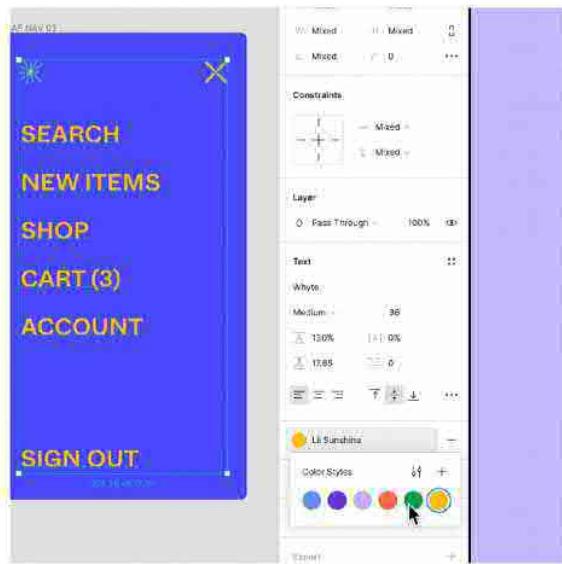
Deskripsi sebuah fitur terkadang dapat menjadi sangat sederhana. Sebuah deskripsi fitur dapat mengacu pada satu atau lebih *user story* (Sommerville, 2021). Pada penelitian ini akan menggunakan Model *input/action/output* dengan menambahkan constraints secara implisit pada komponen terkait saat diperlukan.

2.9 Figma

Figma adalah sebuah alat desain yang mengkombinasikan aksesibilitas dari website dengan fungsionalitas dari aplikasi *native*. Figma memungkinkan kolaborasi *real-time*, menyambut lebih banyak orang ke dalam proses desain, alur kerja yang lebih cepat dan lebih menyenangkan, dan menjadi sebuah rumah untuk bekerja bersama di luar tembok perusahaan. Visi dari Figma adalah membuat desain dapat diakses oleh semua orang (Figma, 2022a). Beberapa fitur dari Figma antara lain Figma Design, Figma Prototyping, dan Figma Design Systems.

Figma Design adalah sebuah fitur untuk membuat sebuah desain dengan disertai alat-alat canggih, dan alur kerja yang lebih efisien. Beberapa keunggulan dari Figma Design antara lain desain untuk web menjadi mudah, desain yang responsif dengan tata letak otomatis, otomatisasi dan peningkatan pekerjaan dengan *plugins* dan *widgets*, efisien, konsisten, dan efisien secara konsisten, dan

mengubah *handoff* pengembang menjadi *handshake*. Desain untuk web menjadi mudah direpresentasikan oleh adanya fitur *modern pen tool*, desain *arc* instan, dan *OpenType*. Desain yang responsif dengan tata letak otomatis direpresentasikan oleh pengubahan ukuran manual yang dikurangi, fitur *stretch to fill*, dan desain yang menggambarkan pengembangan. Otomatisasi dan peningkatan pekerjaan dengan *plugins* dan *widgets* direpresentasikan oleh adanya fitur *plugins*, *widgets*, pengembangan yang mudah, dan *private extension*. Efisien, konsisten, dan efisien secara konsisten direpresentasikan oleh gaya yang fleksibel, desain secara sistematis, dan *libraries* yang dapat diakses. Mengubah *handoff* pengembang menjadi *handshake* direpresentasikan oleh kemampuan *unlimited viewers*, fitur *inspect* file desain, dan ekspor yang mudah (Figma, 2022b).



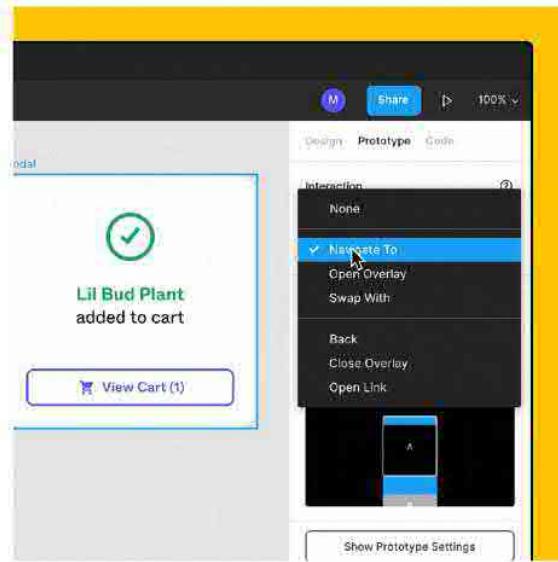
Gambar 2.17 Figma Design yang konsisten, efisien, dan efisien secara konsisten

Sumber: Figma (2022b)

Pada gambar 2.17 terdapat beberapa pengaturan gaya yang dapat digunakan dalam mendesain, antara lain warna, teks, dan pemasian komponen.

Figma Prototyping adalah sebuah fitur untuk membuat *prototype* animasi sebagai *blueprint* yang lebih baik untuk pengembangan. Beberapa keunggulan dari Figma Prototyping antara lain membuat *prototype* yang terasa seperti pengalaman nyata, menunjukkan tanpa memberi tahu visi interaksi, dan melakukan semuanya dalam satu alat tunggal. Membuat *prototype* yang terasa seperti pengalaman nyata direpresentasikan oleh pembangunan yang intuitif, interaksi, dan tampilan *mobile*. Menunjukkan tanpa memberi tahu visi interaksi direpresentasikan oleh transisi lanjutan dengan *smart animate*, *overlays* dinamis, dan *animated GIFs*. Melakukan semuanya dalam satu alat tunggal berarti Figma

Prototyping sebagai platform *all-in-one*, *prototype* yang dapat dibagikan, dan adanya fitur komentar tersemat (Figma, 2022c).



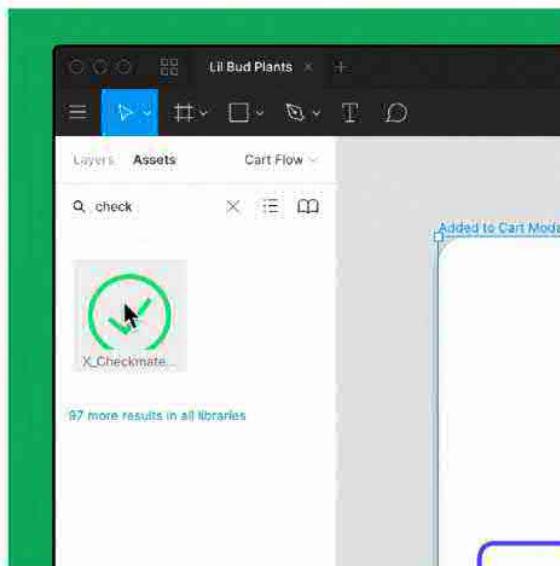
Gambar 2.18 Figma Prototyping membuat *prototype* yang terasa seperti pengalaman nyata

Sumber: Figma (2022c)

Pada gambar 2.18 terdapat beberapa pengaturan *prototype* yang menavigasikan interaksi antar elemen UI.

Figma Design Systems adalah sebuah fitur pada Figma yang akan memperkuat sistem desain, *libraries* yang selalu *up-to-date*, selaras dengan kode, dan mudah dikembangkan seiring perkembangan. Beberapa keunggulan dari Figma Design Systems antara lain semua asset desain terletak pada satu tempat, tempat tim berbagi bahasa yang sama, tempat orang-orang sistem desain menemukan ketenangan pikiran, dan tempat sistem desain bertemu data. Semua asset desain terletak pada satu tempat berarti peningkatan konsistensi desain dengan sistem desain canggih yang dapat diakses seluruh perusahaan, yang mana direpresentasikan oleh asset yang dapat dicari, gaya yang dapat dibagi, dan semuanya pada satu tempat. Tempat tim berbagi bahasa yang sama berarti Figma akan berskala dengan perkembangan sistem desain dan pertumbuhan tim, yang mana direpresentasikan oleh komponen yang dapat digunakan kembali, sesuai dengan kode, dan *copy, code, and go*. Tempat orang-orang sistem desain menemukan ketenangan pikiran berarti Figma akan berskala dengan perkembangan sistem desain dan pertumbuhan tim, yang mana direpresentasikan oleh adanya saring, publikasikan, dan lacak, bangun dengan mempertimbangkan struktur dan skala, dan kontrol izin pengguna. Tempat sistem desain bertemu data berarti adanya *built-in analytics*, yang mana

direpresentasikan oleh pemahaman penggunaan *library*, pembuatan pembaruan yang penting, dan perhitungan dampak desain (Figma, 2022d).



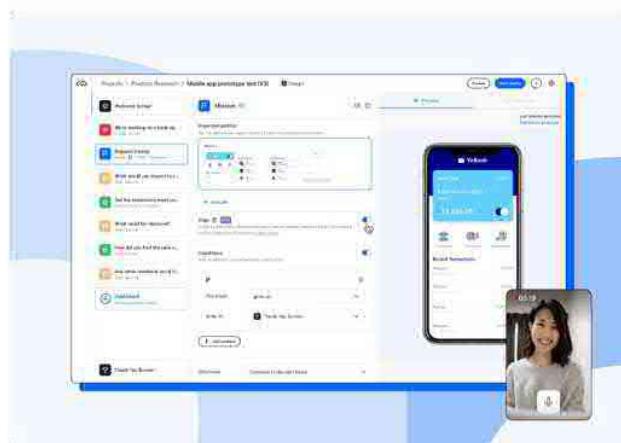
Gambar 2.19 Figma Design Systems yang mana semua asset desain terletak pada satu tempat

Sumber: Figma (2022d)

Pada gambar 2.19 terdapat beberapa asset desain yang terletak pada satu tempat, yaitu pada menu *assets*.

2.10 Maze

Maze adalah sebuah alat untuk melakukan pengujian terhadap sebuah projek, mulai dari *usability test* hingga *survey* pengguna di seluruh dunia hanya dengan menggunakan alamat website dalam hitungan menit. Dengan Maze, wawasan pengguna bisa didapatkan hanya dalam hitungan jam, bukan hari. Maze dapat membantu menciptakan produk yang benar-benar berpusat pada pelanggan, dengan memberdayakan seluruh tim produk untuk mengumpulkan wawasan yang dapat ditindaklanjuti kapan saja dan dimana saja. Maze telah terintegrasi dengan beberapa alat desain seperti Figma, InVision, Adobe XD, Marvel, dan Sketch. Salah satu fitur dari Maze adalah untuk pengujian *prototype*. Pengujian *prototype* digunakan untuk melakukan validasi terhadap desain dan *usability test* secara efisien dalam skala besar (Maze, 2022).



Gambar 2.20 Pengujian *prototype* menggunakan Maze

Sumber: Maze (2022)

Pada gambar 2.20 terdapat pengujian terhadap sebuah *prototype* menggunakan Maze, dimana pada sebelah kiri terdapat sebuah task, pada bagian tengah terdapat detail dari task, dan sebelah kanan terdapat *prototype* yang ingin diuji.

53

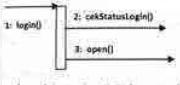
2.11 Unified Modelling Language (UML)

Unified modelling language atau dapat disingkat menjadi UML, merupakan sebuah cara pemodelan secara visual dalam merancang perangkat lunak berorientasi objek. UML juga dapat dijadikan sebuah standar dalam penulisan *blueprint* sebuah sistem, yang termasuk di dalamnya konsep proses bisnis, penulisan kelas dalam sebuah bahasa program spesifik, skema dari *database*, dan komponen-komponen lain yang dibutuhkan dalam sebuah sistem perangkat lunak (Ardonis, 2020). Beberapa diagram yang digunakan pada UML antara lain *sequence diagram*, dan *class diagram*.

2.11.1 Sequence Diagram

Sequence diagram merupakan sebuah diagram yang digunakan dalam penggambaran perilaku dari sebuah objek dengan memberikan peran dan pesan yang saling dikirim dan diterima antar objek (Rosa, & Shalahuddin, 2018). *Sequence diagram* digunakan untuk memodelkan urutan kejadian yang terjadi pada satu eksekusi sistem. Objek pada *sequence diagram* ditunjukkan sebagai sebuah garis vertikal, dan tiap kejadian sebagai sebuah panah horizontal dari objek pengirim kepada objek penerima (Nurralita, 2018).

Berikut merupakan simbol-simbol yang terdapat pada *sequence diagram*.

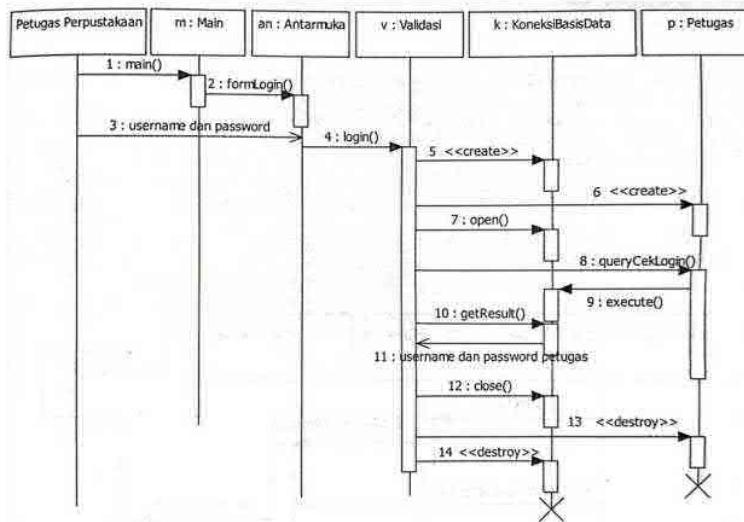
Simbol	Deskripsi
Aktor  atmu  tanpa waktu aktif	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dituliskan menggunakan kata benda di awal frase nama aktor
Garis hidup / lifeline 	menyatakan kehidupan suatu objek
Objek 	menyatakan objek yang berinteraksi pesan
Waktu aktif 	menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  maka cekStatusLogin() dan open() dilakukan di dalam metode login() Aktor tidak memiliki waktu aktif
Pesan tipe create 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe call 	menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
Pesan tipe send 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe return 	menyatakan bahwa suatu objek yang telah menjalankan satu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe destroy 	menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy

Gambar 2.21 Simbol-simbol dalam *sequence diagram*

Sumber: Rosa & Shalahuddin (2018)

Pada gambar 2.21 terdapat beberapa simbol yang digunakan dalam *sequence diagram* beserta penjelasan dan gambarnya. Aktor menggambarkan sebuah entitas yang mana dapat berupa orang, proses, atau sistem lain yang melakukan

interaksi dengan sistem. Garis hidup atau *lifeline* digunakan untuk menyatakan kehidupan sebuah objek. Objek menggambarkan objek yang saling melakukan interaksi melalui pesan. Waktu aktif menggambarkan keadaan aktif dan interaktif sebuah objek dan tahapan yang dilakukan di dalamnya. Pesan tipe *create* menggambarkan sebuah objek yang membuat objek lain, dimana arah panah mengarah kepada objek yang dibuat. Pesan tipe *call* menggambarkan sebuah objek memanggil sebuah metode. Pesan tipe *send* menggambarkan sebuah objek mengirimkan suatu data, masukan, maupun informasi ke objek lain, dimana arah panah mengarah kepada objek yang menerima kiriman. Pesan tipe *return* menggambarkan sebuah objek yang telah menjalankan sebuah metode dan menghasilkan sebuah kembalian lalu mengembalikannya ke objek tertentu, dengan arah panah mengarah kepada objek yang menerima kembalian. Pesan tipe *destroy* menggambarkan sebuah objek yang mengakhiri objek lain, dimana anak panah mengarah kepada objek yang diakhiri.



Gambar 2.22 Contoh sequence diagram

² Sumber: Rosa & Shalahuddin (2018)

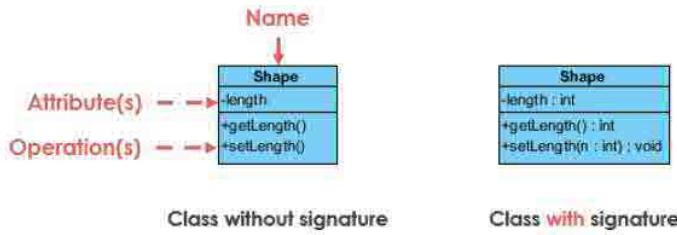
Pada gambar 2.22 terdapat sebuah contoh sequence diagram dari sebuah proses *login*.

Penomoran pada pesan didasarkan pada urutan interaksi yang terjadi. Penggambaran letak pesan haruslah berurutan dari atas ke bawah sesuai pesan yang lebih dahulu berjalan. Semua metode yang terdapat dalam kelas harus ada pada *sequence diagram*, jika tidak maka perancangan metode pada kelas tersebut kurang baik. Hal ini karena adanya metode yang kegunaannya tidak dapat dipertanggungjawabkan (Rosa, & Shalahuddin, 2018).

2.11.2 Class Diagram

Class diagram atau *diagram kelas* merupakan sebuah diagram yang digunakan dalam menggambarkan struktur sebuah sistem dari sisi pendefinisian kelas-kelas yang membangun sistem. Sebuah kelas terdiri atas atribut dan operasi. Atribut adalah variabel-variabel yang terdapat pada suatu kelas, sedangkan operasi adalah fungsi-fungsi yang terdapat pada suatu kelas (Rosa, & Shalahuddin, 2018). *Class diagram* juga menggambarkan relasi antar kelas pada perangkat lunak. *Class diagram* digunakan sebagai acuan pengembang dalam mengembangkan aplikasi agar dokumentasi perangkat lunak sesuai dengan implementasinya (Muchlison, 2021). *Class diagram* pada UML merupakan sebuah diagram yang menggambarkan struktur sistem yang berisikan kelas-kelas, atributnya, operasi, dan relasi antar objek (VisualParadigm, 2022).

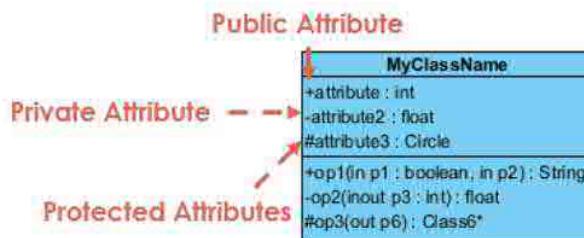
Sebuah kelas pada *class diagram* dapat memiliki *signature* maupun tidak. *Signature* pada atribut diwakili oleh tipe, sedangkan pada operasi diwakili oleh tipe kembalian dari sebuah operasi. Nama kelas merupakan satu-satunya hal yang wajib terdapat pada *class diagram* (VisualParadigm, 2022).



Gambar 2.23 Notasi pada *class diagram*

Sumber: VisualParadigm (2022)

Pada gambar 2.23 terdapat notasi pada *class diagram* dimana terdiri dari nama kelas, atribut, dan operasi. Nama kelas terdapat pada partisi pertama. Atribut terdapat pada partisi kedua. Tipe atribut ditampilkan setelah titik dua. Atribut memetakan variabel sebagai anggota data pada kode. Operasi ditampilkan pada partisi ketiga. Tipe kembalian dari sebuah operasi ditampilkan setelah titik dua terakhir dari nama operasi. Tipe dari parameter ditampilkan setelah titik dua setelah nama parameter tersebut. Operasi memetakan metode ke kelas pada kode (VisualParadigm, 2022).



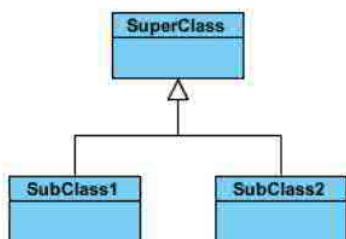
Gambar 2.24 Visibilitas pada *class diagram*

Sumber: VisualParadigm (2022)

Pada gambar 2.24 terdapat definisi dari simbol yang terletak sebelum sebuah atribut, maupun operasi dari suatu kelas yang menunjukkan visibilitas atribut maupun operasi tersebut. Simbol ‘+’ menggambarkan atribut dan operasi *public*, ‘-’ menggambarkan atribut dan operasi *private*, dan ‘#’ menggambarkan atribut dan operasi *protected*.

Sebuah kelas dapat terlibat dengan satu atau lebih relasi atau hubungan dengan kelas lain. Beberapa relasi pada *class diagram* antara lain generalisasi, asosiasi, agregasi, komposisi, dan dependensi (VisualParadigm, 2022).

Generalisasi merupakan relasi dari *kelas yang lebih umum* ke *kelas yang lebih spesifik*, dimana kelas yang lebih spesifik akan mewarisi fitur dari kelas yang lebih umum (VisualParadigm, 2022).



Gambar 2.25 Contoh generalisasi pada *class diagram*

Sumber: VisualParadigm (2022)

Pada gambar 2.25 terdapat sebuah contoh generalisasi pada *class diagram* dimana kelas **SubClass1** dan kelas **SubClass2** mewarisi kelas **SuperClass**.

Asosiasi merupakan relasi struktural antara 2 kelas yang saling berhubungan (VisualParadigm, 2022).

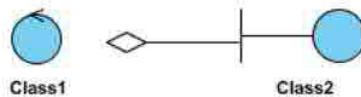


Gambar 2.26 Contoh asosiasi pada *class diagram*

Sumber: VisualParadigm (2022)

Pada gambar 2.26 terdapat sebuah contoh asosiasi pada *class diagram* dimana ada asosiasi yang menghubungkan kelas Class1 dengan kelas Class2.

Agregasi merupakan relasi asosiasi yang lebih khusus, yang mana merupakan relasi struktural antara 2 kelas dimana satu kelas merupakan bagian dari kelas lainnya. Kedua kelas tersebut memiliki masa hidup yang terpisah (VisualParadigm, 2022).



Gambar 2.27 Contoh agregasi pada class diagram

Sumber: VisualParadigm (2022)

Pada gambar 2.27 terdapat sebuah contoh agregasi pada *class diagram* dimana kelas Class2 merupakan agregasi dari kelas Class1. Kelas Class1 dapat memiliki banyak kelas Class2 dengan masa hidup yang saling terpisah.

Komposisi merupakan relasi agregasi yang lebih khusus, yang mana merupakan relasi struktural antara 2 kelas dimana satu kelas merupakan bagian dari kelas lainnya, dan apabila kelas yang merupakan bagian akan ikut hancur apabila kelas utamanya dihancurkan. Masa hidup kelas bagian terikat dengan kelas utama, dan kelas bagian tidak dapat berdiri sendiri (VisualParadigm, 2022).

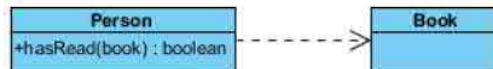


Gambar 2.28 Contoh komposisi pada class diagram

Sumber: VisualParadigm (2022)

Pada gambar 2.28 terdapat sebuah contoh komposisi pada *class diagram* dimana kelas Class2 merupakan komposisi dari kelas Class1. Kelas Class2 akan ikut hancur apabila kelas Class1 dihancurkan, dan kelas Class2 tidak dapat berdiri sendiri karena ia tidak akan ada jika kelas Class1 tidak ada.

Dependensi atau ketergantungan merupakan relasi asosiasi yang lebih khusus, yang mana merupakan relasi struktural antara 2 kelas dimana objek dari satu kelas mungkin menggunakan objek dari kelas lain. Kelas yang bergantung dapat ikut berubah apabila kelas utamanya berubah, namun tidak sebaliknya (VisualParadigm, 2022).



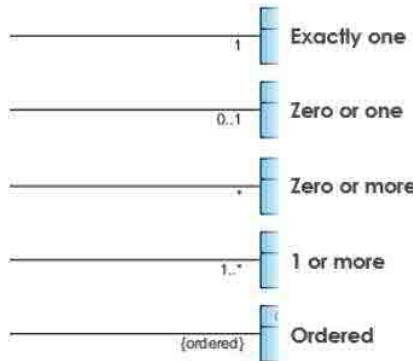
Gambar 2.29 Contoh dependensi pada class diagram

Sumber: VisualParadigm (2022)

Pada gambar 2.29 terdapat sebuah contoh dependensi pada *class diagram* dimana kelas Class2 merupakan dependensi dari kelas Class1. Kelas Class2

bergantung kepada kelas Class1 dimana apabila objek pada kelas Class1 yang dipakai oleh kelas Class2, maka mungkin kelas Class2 juga akan berubah.

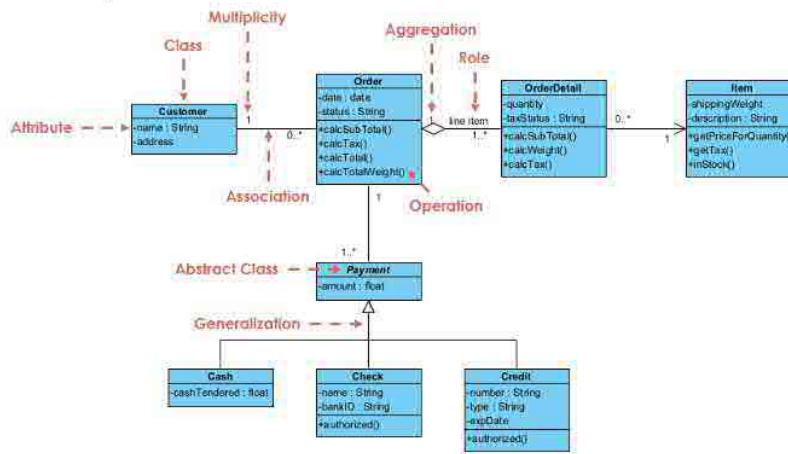
19 Kardinalitas terdiri dari relasi one-to-one atau satu ke satu, relasi one-to-many atau satu ke banyak, dan relasi many-to-many atau banyak ke banyak (VisualParadigm, 2022).



Gambar 2.30 Kardinalitas pada class diagram

Sumber: VisualParadigm (2022)

Pada gambar 2.30 terdapat definisi kardinalitas atau penentuan banyaknya relasi antar objek.



Gambar 2.31 Contoh class diagram

Sumber: VisualParadigm (2022)

20 Pada gambar 2.31 terdapat sebuah contoh *class diagram* yang menunjukkan hubungan antar *kelas* dalam sebuah sistem.

2.12 Pseudocode

Pseudocode merupakan sebuah bahasa penggambaran algoritme dalam penyelesaian sebuah permasalahan, tanpa harus memikirkan pemilihan bahasa pemrograman dan sintaks yang akan digunakan. *Pseudocode* berfungsi mempermudah programmer dalam menulis kode dan memahami algoritme dari kode tersebut (Muchlison, 2021). Berikut merupakan contoh dari sebuah pseudocode.

```
Input: eventUID
output: Unit

Start
    set function to Asynchronous function
    input event collection, eventUID to deleteDocument function
    try
        remove event id from user myEvent field
        call delete file method
        emit FirebaseResponse with Success response
    catch
        emit error message response to FirebaseResponse
End
```

Gambar 2.32 Contoh pseudocode

Sumber: Muchlison (2021)

Pada gambar 2.32 terdapat sebuah contoh pseudocode menghapus event yang berfungsi menghapus sebuah event, dimana berisi input atau masukan, output atau hasil keluaran, start atau mulai dari kode, dan end atau akhir dari kode.

2.13 Aplikasi Perangkat Bergerak

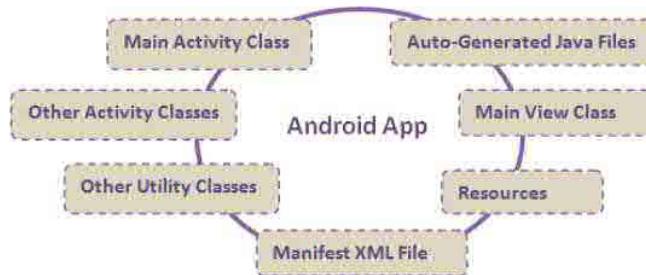
35

Aplikasi Perangkat Bergerak atau aplikasi mobile adalah suatu aplikasi yang memungkinkan digunakan secara mobilitas menggunakan perlengkapan seperti handphone, atau telepon seluler. Dengan menggunakan aplikasi perangkat bergerak, berbagai aktivitas seperti belajar, hiburan, berjualan, bekerja, dan lain sebagainya, dapat dengan mudah dilakukan. Secara keseluruhan, aplikasi perangkat bergerak adalah sebuah aplikasi yang dapat dipakai meskipun pengguna berpindah tempat tanpa terputusnya komunikasi (Muradi, 2021).

Salah satu bentuk aplikasi perangkat bergerak adalah aplikasi Android. Pemilihan Android sebagai bentuk aplikasi perangkat bergerak dalam penelitian ini dimaksudkan untuk menjangkau pengguna yang lebih banyak, dan didasari statistik mengenai jumlah pengguna sistem operasi Android di Indonesia yang mencapai 90,8% (Statcounter, 2022).

2.14 Android

Android adalah sistem operasi perangkat bergerak yang mengadopsi sistem operasi Linux yang telah dimodifikasi. Android merupakan sistem operasi yang menyediakan *platform* bagi para pengembang secara terbuka untuk membuat aplikasinya sendiri. Kemudahan dalam memenuhi kebutuhan secara praktis dan cepat merupakan salah satu hal yang disediakan aplikasi Android (Agustina, & Suprianto, 2012).



Gambar 2.33 Komponen aplikasi Android

Sumber: Abdelbaki, Charkaoui, & Lahmar (2016)

Pada gambar 2.33 terdapat komponen yang terdapat pada arsitektur aplikasi Android. Arsitektur aplikasi Android didasarkan pada Runtime, kernel Linux dan satu set *library* yang dapat diakses melalui kerangka kerja aplikatif. Kerangka Aplikatif menyediakan layanan dalam bentuk kelas java untuk mesin runtime dan aplikasi, dan manajemennya. Aplikasi Android ditulis dalam Java, tetapi dijalankan oleh mesin virtual tertentu yang disebut Dalvik. Mesin eksekusi Dalvik dan Android bergantung pada kernel Linux yang menangani interaksi dengan perangkat keras (driver dan manajemen memori). Sementara itu, satu set *API* menyediakan akses ke semua layanan, fungsi, dan peralatan. Pada Android, terdapat beberapa fragmentasi fungsi seperti penyimpanan, akses ke data ponsel, komunikasi antar aplikasi, akses ke fitur ponsel, keamanan dan proteksi, dan antarmuka pengguna (Abdelbaki, et al., 2016).

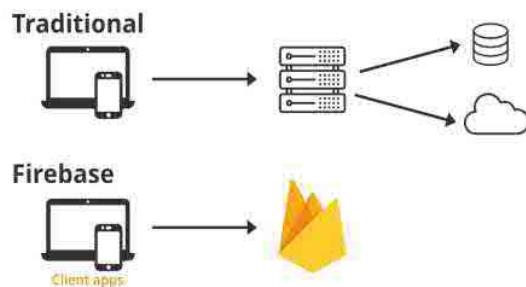
2.15 Android Studio

Android Studio adalah *Integrated Development Environment* berbasis IntelliJ IDEA dan *Gradle* yang dibuat oleh Google dan dikhawasukan untuk pengembangan aplikasi Android. Android Studio menawarkan ketabilan lebih pada pengembangan aplikasi Android, pembuatan *realtime layout* dengan *preview*, dan kemudahan dalam penggunaan *library* tambahan (Putra, 2017).

2.16 Firebase

Firebase adalah sebuah BaaS atau *Back-end as a Service* yang dimiliki dan ditawarkan oleh Google untuk mempermudah pengembang perangkat lunak perangkat bergerak. Dengan adanya layanan Firebase, pengembang aplikasi

dapat lebih fokus mengembangkan aplikasinya tanpa harus memberikan usaha lebih dalam urusan *back-end* (Nugraha, 2019). Tidak diperlukan pemrograman dari sisi server Firebase membuat fitur-fiturnya mudah digunakan secara lebih efisien. Layanan Firebase merupakan layanan penyimpanan awan dan dapat digunakan untuk Android, ios, website, dan Unity (GeeksForGeeks, 2021). Firebase sendiri menyediakan beberapa layanan, antara lain Firebase Authentication, Firebase Cloud Firestore, dan Firebase Cloud Storage (Firebase, 2022a).



Gambar 2.34 Skema arsitektur *back-end* Firebase dibanding *back-end* tradisional

Sumber: GeeksForGeeks (2021)

Pada gambar 2.34 terdapat perbandingan arsitektur *back-end* yang terdapat pada Firebase dan arsitektur *back-end* tradisional. Pada *back-end* tradisional, data pada aplikasi akan diambil dari awan atau internet dan dari *database*. Sedangkan, pada *back-end* Firebase kedua sumber data tersebut digantikan oleh hanya Firebase.

2.16.1 Firebase Authentication

Firebase Authentication adalah sebuah layanan sistem autentikasi yang aman, sekaligus meningkatkan pengalaman autentikasi dan orientasi pengguna. Layanan ini mendukung autentikasi menggunakan *email* dan *password*, autentikasi telepon, Google, Twitter, Facebook, GitHub, dan lainnya. Firebase Authentication dapat menghemat waktu berbulan-bulan dalam menyiapkan sistem autentikasi sendiri, yaitu hanya dalam kode yang jumlahnya di bawah 10 baris, bahkan dengan kasus kompleks seperti penggabungan akun (Firebase, 2022b).

Tabel 2.2 Dependensi Firebase Authentication

```
dependencies {
    // Impor BoM untuk platform Firebase
    implementation platform('com.google.firebaseio:firebase-bom:30.3.1')

    // Deklarasikan dependensi untuk library Firebase Authentication
    // Saat menggunakan BoM, Anda tidak menentukan versi di dependensi
    library Firebase
    implementation 'com.google.firebaseio:firebase-auth-ktx'
}
```

Sumber: Firebase (2022c)

Pada tabel 2.2 terdapat kode penambahan dependensi pada Gradle yang harus ditambahkan agar dapat menggunakan Firebase Authentication.

2.16.2 Firebase Cloud Firestore

Firebase Cloud Firestore adalah sebuah database *NoSQL* berbasis dokumen yang memungkinkan menyimpan, menyinkronkan, dan membuat kueri data untuk aplikasi perangkat bergerak dalam skala global dengan mudah. Data disusun dalam bentuk koleksi dan dokumen berhierarki (Firebase, 2022d).

Tabel 2.3 Dependensi Firebase Cloud Firestore

```
dependencies {
    // Impor BoM untuk platform Firebase
    implementation platform('com.google.firebaseio:firebase-bom:30.3.1')

    // Deklarasikan dependensi untuk library Firebase Authentication
    // Saat menggunakan BoM, Anda tidak menentukan versi di dependensi
    library Firebase
    implementation 'com.google.firebaseio:firebase-firestore'
}
```

Sumber: Firebase (2022e)

Pada tabel 2.3 terdapat kode penambahan dependensi pada Gradle yang harus ditambahkan agar dapat menggunakan Firebase Cloud Firestore.

46

2.16.3 Firebase Cloud Storage

Firebase Cloud Storage adalah sebuah layanan penyimpanan yang dapat menyimpan dan menyajikan konten buatan pengguna dengan mudah. Firebase Cloud Storage dapat dikembangkan dengan mudah dari prototipe ke produksi menggunakan teknologi yang sama dengan yang dipakai aplikasi Spotify dan Google Photo (Firebase, 2022f).

Tabel 2.4 Dependensi Firebase Cloud Storage

```
dependencies {
    // Impor BoM untuk platform Firebase
    implementation platform('com.google.firebaseio:firebase-bom:30.3.1')

    // Deklarasikan dependensi untuk library Firebase Authentication
    // Saat menggunakan BoM, Anda tidak menentukan versi di dependensi
    library Firebase
    implementation 'com.google.firebaseio:firebase-storage-ktx'
}
```

Sumber: Firebase (2022g)

Pada tabel 2.4 terdapat kode penambahan dependensi pada Gradle yang harus ditambahkan agar dapat menggunakan Firebase Cloud Storage.

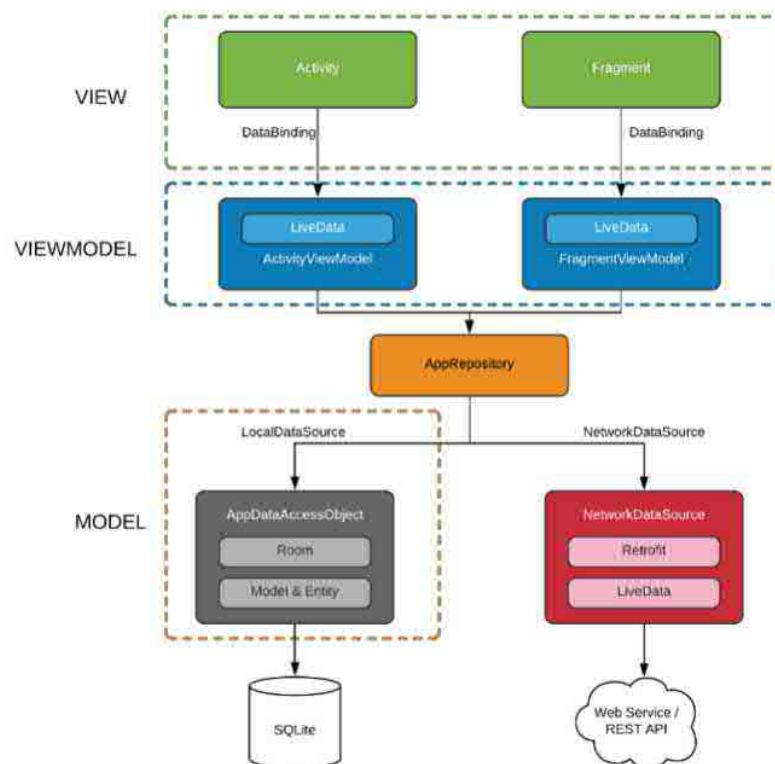
2.17 Arsitektur Pengembangan Aplikasi Android

Tuntutan bisnis yang lebih baru membuat sistem perangkat lunak menjadi sangat rumit dan canggih. Oleh karena itu, arsitektur perangkat lunak sangat cocok dalam mengatasi peningkatan kompleksitas dan perubahan perangkat lunak (Raj, Raman, & Subramanian, 2017). Pola arsitektur adalah sebuah pola yang terkenal dalam penyelesaian permasalahan arsitektur perangkat lunak. Pola arsitektur dari sebuah perangkat lunak merupakan keseluruhan ‘organisasi’ kode (Arif, Musthafa, & Muriyatmoko, 2019).

Model-View-ViewModel atau MVVM adalah salah satu pola arsitektur yang mengusung pemisahan presentasi, yang mana memisahkan antara grafis antarmuka dan proses logika bisnis (Arif, Musthafa, & Muriyatmoko, 2019). Beberapa komponen dari pola MVVM yaitu Model, View, dan ViewModel. Model mewakili proses logika bisnis dan data, yang mana dalam hal ini proses bisnis logika menentukan bagaimana data dimanipulasi. View mewakili komponen grafis antarmuka, namun hanya bertanggung jawab mewakili data, tanpa memanipulasinya. Pada MVVM, View merupakan komponen aktif yang berisi perilaku, peristiwa, dan pengikatan data, yang mana membutuhkan informasi mengenai komponen ViewModel, dan Model yang mendasarinya. ViewModel adalah sebuah komponen yang membantu pemisahan presentasi, dalam hal ini memisahkan View dan Model, sekaligus sebagai pengontrol yang mendukung koordinasi dan interaksi antara keduanya. ViewModel juga membantu mempertahankan state tampilan, memanipulasi Model yang dilakukan pada View, dan memicu peristiwa dalam View itu sendiri (Raj, Raman, & Subramanian, 2017).

Pemilihan pola MVVM pada pengembangan penelitian ini berdasarkan beberapa keuntungan pada pola MVVM itu sendiri. Keuntungan-keuntungan tersebut yaitu *Maintainability*, *Testability*, dan *Extensibility*. Pada sisi *Maintainability* atau pemeliharaan, pemisahan potongan kode yang berbeda

membuatnya lebih mudah dalam mengelola kode, dan memastikan rilis kode yang lebih cepat. Pada sisi *testability* atau kemampuan untuk diuji, pemisahan potongan kode yang berbeda sangat granular, dan merupakan sebuah kunci inti logika fungsional, yang membuat pengujian unit lebih mudah. Pada sisi *extensibility* atau kemampuan ekstensi, potongan kode-kode yang granular memungkinkan penggunaan kembali kode, dan modifikasi cepat dari cuplikan kode. Pola ini juga sangat cocok untuk aplikasi yang membutuhkan pengujian unit menyeluruh, pengembangan yang menggunakan konsep kode yang dapat digunakan kembali, dan fleksibilitas dalam mengubah antarmuka pengguna tanpa mengubah basis kode (Raj, Raman, & Subramanian, 2017).



Gambar 2.35 Rancangan arsitektur MVVM

Sumber: Arif, Musthafa, & Muriyatmoko (2019)

Pada gambar 2.35 terdapat salah satu contoh rancangan arsitektur MVVM yang mana terdiri atas Model, View, dan ViewModel. Model sendiri direpresentasikan oleh LocalDataSource, yang mana berasal dari AppDataAccessObject, Room, dan Model & Entity, dan mengambil dan menyimpan data dari SQLite. Terdapat juga NetworkDataSource, yang mana berasal dari NetworkDataSource, Retrofit, dan LiveData, dan mengambil data dari web service maupun REST API. Kedua LocalDataSource dan NetworkDataSource merupakan sumber dari AppRepository yang nantinya akan

mengatur dan memberikan alur data ke ViewModel, maupun sebaliknya. ViewModel direpresentasikan oleh ViewModel yang berisi LiveData, yang nantinya akan memberikan data secara dinamis untuk ditampilkan pada View. View direpresentasikan oleh kelas tampilan seperti Activity, maupun Fragment, yang akan melakukan *databinding* atau pengikatan data ke ViewModel terkait.

2

2.18 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah aktivitas yang memiliki tujuan mengevaluasi kemampuan sebuah sistem atau program, dan menentukan kesesuaian dengan hasil yang diharapkan. Pengujian perangkat lunak secara spesifik dapat diartikan sebagai proses eksekusi sebuah program guna menemukan *bug* atau kesalahan. Pengujian perangkat lunak dapat dikatakan sukses apabila sebuah kesalahan dapat ditemukan yang awalnya tak ditemukan (Dahono, 2020). Sebuah pengembangan perangkat lunak memiliki sebuah pengujian yang digunakan untuk mengetahui tingkat keberhasilan pengembangan perangkat lunak. Tujuan dari pengujian perangkat lunak adalah untuk mengetahui tingkat ketepatan fungsi dan kelayakan dari perangkat lunak tersebut (Muchlison, 2021).

Fitur dari sebuah produk mengimplementasikan beberapa fungsionalitas pengguna yang berguna. Untuk mengetahui apakah fitur diimplementasikan seperti yang diharapkan dan memenuhi kebutuhan pengguna, harus dilakukan pengujian terhadap fitur. Cara yang baik untuk melakukan pengujian terhadap sebuah fitur adalah seputar *scenario* atau serangkaian *user story* (Sommerville, 2021).

Story title	User story
User registration	As a user, I want to be able to log in without creating a new account so that I don't have to remember another login ID and password.
Information sharing	As a user, I want to know what information you will share with other companies. I want to be able to cancel my registration if I don't want to share this information.
Email choice	As a user, I want to be able to choose the types of email that I'll get from you when I register for an account.

Gambar 2.36 Contoh *user story* untuk fitur masuk dengan Google

Sumber: Sommerville (2021)

Pada gambar 2.36 terdapat *user story* untuk fitur masuk dengan Google.

Test	Description
Initial login screen	Test that the screen displaying a request for Google account credentials is correctly displayed when a user clicks on the "Sign-in with Google" link. Test that the login is completed if the user is already logged in to Google.
Incorrect credentials	Test that the error message and retry screen are displayed if the user inputs incorrect Google credentials.
Shared information	Test that the information shared with Google is displayed, along with a cancel or confirm option. Test that the registration is canceled if the cancel option is chosen.
Email opt-in	Test that the user is offered a menu of options for email information and can choose multiple items to opt in to emails. Test that the user is not registered for any emails if no options are selected.

Gambar 2.37 Pengujian fitur masuk dengan Google

Sumber: Sommerville (2021)

Pada gambar 2.37 terdapat sebuah contoh pengujian fitur masuk dengan Google.

Misalkan, pada sebuah fitur masuk dengan Google tercakup dalam 3 *user story* seperti pada gambar di atas. Berdasarkan *user story* yang ada, dapat dikembangkan serangkaian pengujian untuk fitur tersebut (Sommerville, 2021).

2.18.1 Black Box Testing

Black box testing adalah sebuah pendekatan pengujian dimana pengujian dilakukan dengan membuat kasus uji terhadap setiap fungsi. *Black box testing* digunakan untuk memastikan keselarasan semua alur perangkat lunak dengan kebutuhan pengguna. Pengujian ini dilakukan berdasarkan kebutuhan fungsional yang ada pada sistem (Muchlison, 2021). *Black box testing* bertujuan untuk mengetahui apakah program tersebut sesuai dengan tugas yang diharapkan dari program tanpa mengetahui kodennya. Pengujian ini dilakukan dengan mengidentifikasi masukan, dan mengujinya agar mengetahui letak kesalahannya (Ningrum, et al., 2019).

Id	Deskripsi Pengujian	Hasil yang Diharapkan
L01	Mengisi Username dengan "dandi123", Password diisi dengan "dan123" dengan ketentuan data yang diisi merupakan data yang ada di database kemudian klik tombol Login.	Akses Login berhasil, sistem akan menutup form Login dan menampilkan form Data.
L02	Mengisi Username dengan "abc123", Password diisi dengan "abc123" dengan ketentuan data yang diisi merupakan data yang tidak ada atau belum tersimpan di database, kemudian klik tombol Login.	Akses Login Gagal, sistem tidak dapat membuka form Data dan tampilan tetap pada form Login.

Gambar 2.38 Contoh rancangan kasus uji form login menggunakan black box testing

Sumber: Ningrum, et al. (2019)

Pada gambar 2.38 terdapat sebuah contoh rancangan kasus uji form *login* menggunakan *black box testing*, dimana terdapat dua kasus uji dengan hasil yang berbeda. Pada pengujian kasus uji pertama, akan dilakukan pengisian *username*

dan *password* yang sesuai dengan yang ada di *database*. Hasil yang diharapkan dari kasus uji pertama yaitu akses *login* yang berhasil, ditutupnya form *login*, dan menampilkan form data. Sedangkan, pada pengujian kasus uji kedua akan dilakukan pengisian *username* dan *password* yang tidak ada atau belum tersimpan pada *database*. Hasil yang diharapkan dari kasus uji kedua yaitu akses *login* yang gagal, tidak terbukanya form data, dan tampilan tetap pada form *login*.

2.18.2 Usability Testing

Usability secara umum mengacu pada sejauh mana pengguna dapat belajar dan menggunakan produk dalam mencapai tujuannya, dan seberapa jauh ke tingkat kepuasan pengguna dalam menggunakan produk. *Usability* merupakan ukuran kualitas dari sebuah sistem yang digunakan untuk menilai seberapa mudah pengguna mengenal *user interface* yang didasarkan pada pengalaman pengguna ketika menggunakan sistem (Pratama, Jonemaro, & Dewi, 2021).

Usability testing adalah sebuah pengujian guna mengetahui apakah sebuah *method*, *class*, sistem, ataupun subsistem, telah memenuhi kebutuhan pengguna. Tujuan dari *usability testing* adalah agar mendapatkan umpan balik secara cepat dalam meningkatkan antarmuka, dan melakukan koreksi terhadap kesalahan dalam komponen perangkat lunak (Dahono, 2020). *Usability testing* menguji apakah pengguna dapat mengerti cara menggunakan sistem secara cepat, tanpa melakukan kesalahan (Sommerville, 2021). Secara umum, *usability testing* dilakukan dengan mengevaluasi produk kepada sekitar 10 pengguna. Jumlah pengguna yang melakukan pengujian bisa lebih atau kurang dari 10, namun 10 merupakan angka standar yang digunakan (LaRoche, 2020).

Salah satu jenis pengujian pada *usability testing* adalah *moderated usability testing*. *Moderated usability testing* adalah semua jenis *usability testing* dimana moderator menjalankan sesi, mengajukan pertanyaan kepada partisipan, dan mengontrol jalannya pengujian. Moderator merupakan seseorang yang berinteraksi dengan partisipan, menanyakan pertanyaan, duduk bersama dengan partisipan saat pengujian, dan mencoba mendapatkan sebanyak mungkin data dari partisipan. Tujuan dari moderator adalah menjaga sesi tetap terkendali, tidak membahayakan melainkan melindungi partisipan, selalu memberitahu partisipan seakan-akan tidak sedang diuji, bersikap profesional, empatis, dan membantu partisipan tanpa membantu mereka dalam pengujian, memimpin partisipan, menunjukkan *vibes* positif, dan membuat partisipan berbicara secara lantang dan belajar tentang mengapa mereka melakukan apa yang mereka lakukan dalam pengujian (LaRoche, 2020).

Salah satu aspek pengukuran dalam *usability testing* yaitu *effectiveness* atau efektivitas. Tingkat efektivitas aplikasi diukur menggunakan *completion rate* atau tingkat keberhasilan. Data yang dibutuhkan dalam pengukuran aspek efektivitas yaitu berhasil atau gagalnya partisipan dalam menjalankan tugas (Wibawa, Mursityo, & Retno, 2019).

$$\text{Effectiveness} = \frac{\text{Number of tasks completed successfully}}{\text{Total number of tasks undertaken}} \times 100\%$$

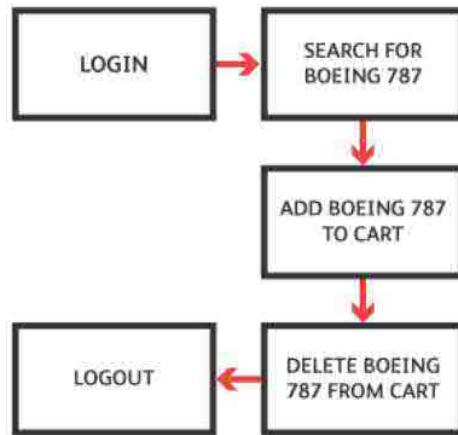
Gambar 2.39 Pengukuran tingkat efektivitas pada *usability testing*

Sumber: Wibawa, Mursityo, & Retno (2019)

Pada gambar 2.39 terdapat rumus pengukuran tingkat efektivitas pada *usability testing*. Untuk tingkat efektivitas dapat dihitung dengan membagi jumlah tugas yang berhasil diselesaikan dengan jumlah tugas yang diberikan, lalu dikali dengan 100%.

2.18.3 Scenario-Based Testing

Scenario atau dapat juga disebut skenario adalah sebuah cerita yang bertujuan memecahkan tugas atau permasalahan yang rumit. Dalam pengujian, skenario adalah sebuah cara pengguna memanfaatkan perangkat lunak dalam kondisi dunia nyata guna memastikan verifikasi *end-to-end* fungsionalitas sistem. *Scenario-based testing* adalah sebuah pengujian yang berbasis skenario yang berfokus pada pengujian alur bisnis perangkat lunak. Tujuannya adalah untuk menemukan *bug* yang tidak akan dideteksi pengujian lain. Skenario di sini menyerupai instruksi terkait bagaimana perangkat lunak digunakan, dan skenario pengujian dapat berupa kasus pengujian terpisah, atau serangkaian kasus-kasus pengujian yang saling terhubung (QATestLab, 2018).



Gambar 2.40 Contoh skenario pada *scenario-based testing*

Sumber: SoapUI (2022)

Pada gambar 2.40 terdapat sebuah contoh skenario pada *scenario-based testing* mulai dari *login*, pencarian Boeing 787, penambahan dan penghapusan Boeing 787 dari keranjang, dan *logout*.

Scenario-based testing harus dikaitkan secara langsung dengan *user story*. Dalam keadaan ideal, pengujian ini harus menjadi *user story* agar kasus pengujian dapat lebih mudah dikembangkan, dan hasil lebih mudah dipahami (SoapUI, 2022).

2.18.4 System Usability Scale

Salah satu teknik pengukuran dalam *usability testing* adalah *system usability scale* atau dapat disingkat SUS. SUS adalah sebuah teknik pengujian dengan cara melibatkan pengguna akhir dalam pengujinya. Instrumen pengujian dalam SUS yaitu 10 pertanyaan sebagai tolak ukur pengujian, dan skala penilaian sebagai ukuran pembobotan pengujian dari 1 hingga 5, dimana 1 menunjukkan sangat tidak setuju, dan 5 menunjukkan sangat setuju terhadap pertanyaan pengujian (Ependi, Kurniawan, & Panjaitan, 2019).

Berikut merupakan instrumen pengujian SUS.

Tabel 2.5 Instrumen pengujian SUS

No	Pertanyaan	Skala
1	Saya ingin lebih sering menggunakan aplikasi ini	1 - 5
2	Saya merasa aplikasi ini tidak seharusnya dibuat serumit ini	1 - 5
3	Saya pikir aplikasi ini mudah digunakan	1 - 5
4	Saya membutuhkan bantuan dari orang teknis untuk menggunakan aplikasi ini	1 - 5
5	Saya menemukan bahwa fitur pada aplikasi ini terintegrasi dengan baik	1 - 5
6	Saya berfikir ada ketidaksesuaian dalam aplikasi ini	1 - 5
7	Saya merasa kebanyakan orang akan mudah mempelajari aplikasi ini dengan sangat cepat	1 - 5
8	Saya menemukan bahwa aplikasi ini sangat rumit untuk digunakan	1 - 5
9	Saya percaya diri untuk menggunakan aplikasi ini	1 - 5
10	Saya perlu belajar sebelum menggunakan aplikasi ini	1 - 5

Sumber: Adaptasi dari Ependi, Kurniawan, & Panjaitan (2019)

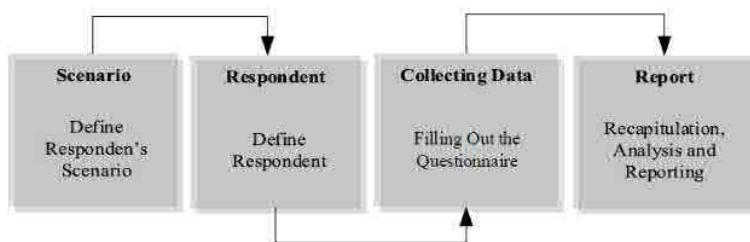
Pada tabel 2.5 terdapat instrumen pengujian berupa 10 pertanyaan yang nantinya akan diajukan kepada pengguna dan kemudian hasilnya akan digunakan sebagai perhitungan hasil pengujian.

Jawaban	Skor
Sangat Tidak Setuju (STS)	1
Tidak Setuju (TS)	2
Ragu-ragu (RG)	3
Setuju (S)	4
Sangat Setuju (SS)	5

Gambar 2.41 Skala penilaian SUS

Sumber: Muchlison (2021)

Pada gambar 2.41 terdapat skala penilaian dalam SUS dari skala 1 sampai 5.

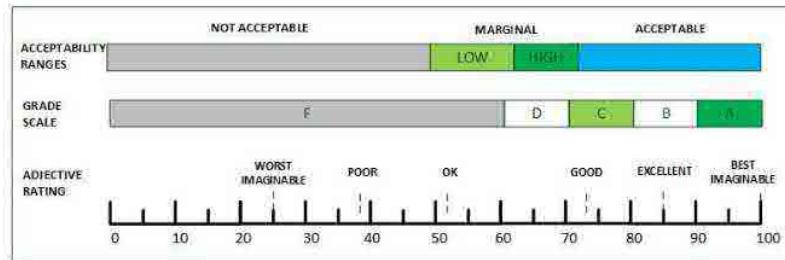


Gambar 2.42 Langkah-langkah SUS

Sumber: Ependi, Kurniawan, & Panjaitan (2019)

Pada gambar 2.42 terdapat langkah-langkah dalam melakukan SUS. Langkah-langkah pengujian dalam SUS dimulai dari penentuan skenario, menentukan responden, melakukan pengujian oleh responden, dan merekapitulasi hasil pengujian.

Cara perhitungan hasil pengujian dalam SUS antara lain pada pertanyaan instrumen dengan nomor ganjil skala jawaban instrumen dikurangi satu, pada pertanyaan instrumen dengan nomor genap memiliki perhitungan angka lima dikurangi skala jawaban instrumen, hasil penilaian skala 0 sampai 4 dengan 4 sebagai jawaban terbaik, penjumlahan jawaban dikali dengan 2,5, dan menentukan nilai rata-rata jawaban instrumen pengujian semua responden. Dalam menentukan hasil perhitungan, terdapat 3 aspek yaitu *acceptability ranges*, *grade scale*, dan *adjective rating*. *Acceptability ranges* digunakan untuk mengetahui tingkat penerimaan pengguna terhadap perangkat lunak. *Grade scale* digunakan untuk mengetahui tingkatan perangkat lunak. *Adjective rating* digunakan untuk melihat *rating* dari perangkat lunak. (Ependi, Kurniawan, & Panjaitan, 2019).



Gambar 2.43 Penilaian SUS

Sumber: Ependi, Kurniawan, & Panjaitan (2019)

Pada gambar 2.43 terdapat penilaian pada SUS yang mengacu pada hasil perhitungan pengujian dalam SUS.

BAB 3 METODOLOGI

3.1 Tipe Penelitian

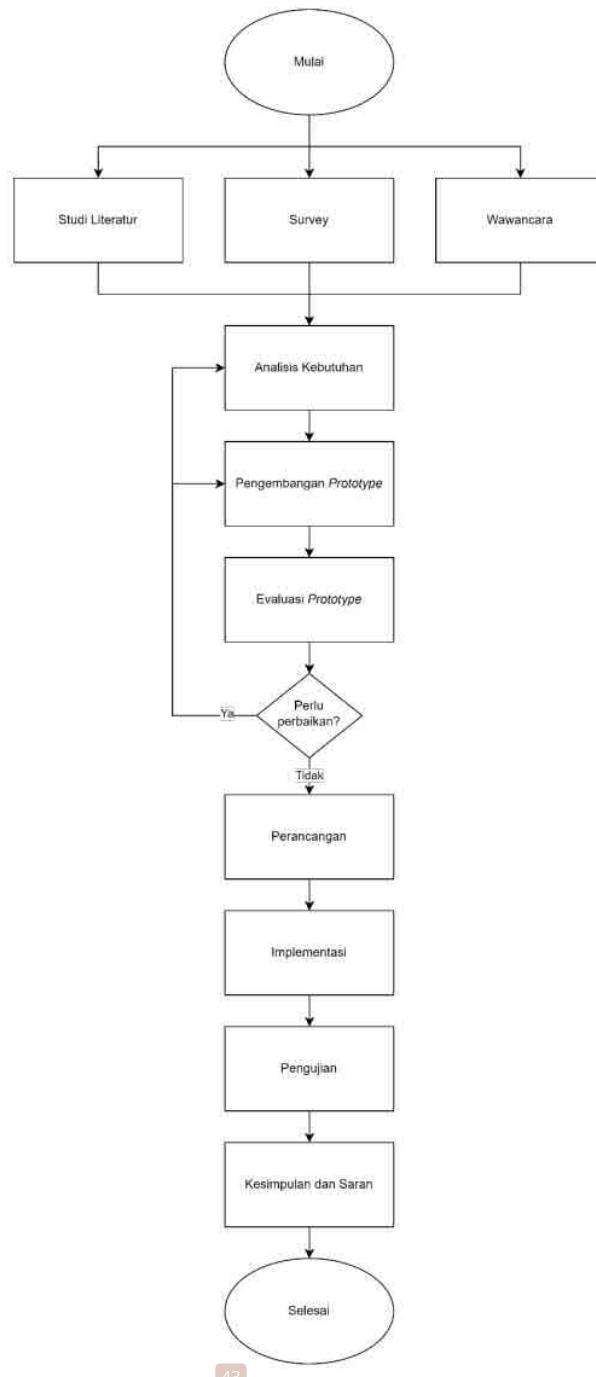
Tipe penelitian adalah sebuah klasifikasi dari jenis penelitian skripsi berdasarkan produk utama yang dihasilkan. Tipe penelitian yang diambil pada penelitian ini yaitu implementatif pengembangan. Hasil artefak dari penelitian ini yaitu sebuah aplikasi perangkat bergerak berbasis Android.

3.2 Metode Penelitian

Pada tipe penelitian implementatif, terdapat metode atau strategi secara umum dalam menyelesaikan permasalahan penelitian. Dalam hal ini, metode penelitian dapat terkait dengan pembuatan artefak, studi kasus, survei, dan lain sebagainya. Adapun metode penelitian ini yaitu sebagai berikut:

1. Studi Literatur, *Survey*, dan Wawancara
2. Analisis Kebutuhan
3. Pengembangan *Prototype*
4. Evaluasi *Prototype*
5. Perancangan
6. Implementasi
7. Pengujian
8. Kesimpulan dan Saran
9. Selesai

Berikut merupakan metode penelitian tersebut jika digambarkan dalam bentuk diagram alur.



43

Gambar 3.1 Metode penelitian

Pada gambar 3.1 terdapat metode penelitian yang digunakan pada penelitian ini dalam bentuk diagram alur. Tahap Studi Literatur digunakan untuk pengumpulan teori pendukung penelitian. Survey, dan Wawancara digunakan sebagai pengumpulan data pendukung penelitian. Analisis kebutuhan digunakan sebagai acuan spesifikasi perangkat lunak yang ingin dibuat, dimana terdiri dari pengembangan *persona*, *scenario*, *user story*, dan fitur. Pada tahap Pengembangan *Prototype*, akan dikembangkan sebuah *prototype* untuk nantinya dievaluasi kepada pengguna pada tahap Evaluasi *Prototype*. Nantinya, hasil dari evaluasi akan dijadikan bahan pertimbangan untuk perlu tidaknya dilakukan perbaikan. Tahap Perancangan akan merancang perangkat lunak yang akan dikembangkan, berdasarkan *prototype* yang telah tervalidasi pada tahap evaluasi. Tahap Implementasi akan mengimplementasikan rancangan perangkat lunak menjadi sebuah perangkat lunak. Perangkat lunak yang telah dibuat selanjutnya akan diuji pada tahap Pengujian, guna mengetahui tingkat ketepatan perangkat lunak yang dikembangkan dengan spesifikasi yang telah ditetapkan, dan sebagai bahan penarikan kesimpulan dan saran.

3.3 Subjek Penelitian

Subjek penelitian untuk pengembangan aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa berbasis Android adalah mahasiswa yang suka, terkadang, atau pernah memasak. Pada penelitian ini, studi kasus yang diambil yaitu terhadap mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya.

3.4 Lokasi Penelitian

Lokasi penelitian ini dilakukan di Fakultas Ilmu Komputer Universitas Brawijaya.

3.5 Teknik Pengumpulan Data

Teknik pengumpulan data yang dipakai dalam penelitian "Pengembangan Aplikasi Perangkat Bergerak Panduan dan Resep Masakan untuk Mahasiswa Berbasis Android" adalah kuesioner via Google Forms, dan wawancara. Pengumpulan data kuesioner dilakukan 2 kali pada tahap pra-penelitian, yaitu kepada 130 masyarakat secara umum, dan secara spesifik kepada 82 mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya. Pada tahap analisis kebutuhan, akan dilakukan wawancara kepada mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya, untuk nantinya akan diobservasi untuk dijadikan acuan dalam pembuatan *persona*, *user story*, *scenario*, dan fitur. Pada tahap evaluasi *prototype*, akan dilakukan pengumpulan data dengan melakukan evaluasi terhadap *prototype* secara daring menggunakan Maze. Terakhir, akan dilakukan pengumpulan data untuk tahap pengujian terhadap aplikasi secara luring.

3.6 Peralatan Pendukung yang Digunakan

Peralatan pendukung yang digunakan pada pengembangan perangkat lunak dalam penelitian ini terdiri dari dua macam peralatan yaitu perangkat lunak dan

perangkat keras. Perangkat lunak yang digunakan yaitu Android Studio untuk pembuatan aplikasi perangkat bergerak Android, dan Figma untuk pembuatan prototype dan desain antarmuka perangkat lunak. Perangkat keras yang digunakan yaitu laptop ASUS f570zd, dan perangkat bergerak Samsung A52s dengan sistem operasi Android 11.

21

3.7 Metode Pengembangan

Metode pengembangan perangkat lunak yang digunakan pada penelitian ini yaitu Prototyping. Metode ini dipilih karena kemampuannya yang memungkinkan perangkat lunak beradaptasi terhadap perubahan kondisi, menambahkan kebutuhan fungsional, mengimplementasikan proses pengembangan secara periodik guna menemukan solusi selama pengembangan, meminimalkan dokumentasi yang harus dibuat pengembang, dan memperbolehkan perubahan terkait kebutuhan, kode, dan desain dari perangkat lunak yang telah dibuat.

3.7.1 Studi Literatur, Survey, Wawancara

2

Studi Literatur dilakukan dengan mengumpulkan teori pendukung penelitian yang bersumber dari literatur, buku, dan referensi yang terkait dengan penelitian ini. Survey dilakukan dengan menggunakan Google Form, yang akan disebar kepada responden. Wawancara dilakukan dengan mewawancarai langsung pengguna, dalam penelitian ini terbatas pada Mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya.

Studi Literatur bertujuan untuk pengumpulan teori pendukung penelitian. Survey, dan wawancara bertujuan untuk pengumpulan data pendukung penelitian. Studi Literatur dilakukan pada tahap awal penelitian ini, dimana telah dijabarkan pada bab landasan kepustakaan. Survey dilakukan pada tahap pra-penelitian, dan tahap pengujian. Survey pada tahap pra-penelitian digunakan sebagai landasan dan latar belakang dari penelitian ini. Survey pada tahap pengujian digunakan untuk mengevaluasi kemampuan sebuah sistem atau program, dan menentukan kesesuaian dengan hasil yang diharapkan. Wawancara dilakukan pada tahap Analisis Kebutuhan, dan saat tahap Evaluasi Prototype. Wawancara pada tahap Analisis Kebutuhan digunakan sebagai landasan, referensi, dan acuan dalam pembuatan persona, scenario, user story, dan fitur. Wawancara pada tahap Evaluasi Prototype digunakan untuk mengevaluasi prototype yang telah dikembangkan.

3.7.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan mengembangkan persona, scenario, user story, dan fitur, yang didasari pada pengumpulan data yang bersumber dari wawancara. Analisis kebutuhan dengan pengembangan persona, scenario, user story, dan fitur bertujuan untuk mengetahui kebutuhan dan spesifikasi perangkat lunak yang ingin dibuat. Dilakukan juga analisis kompetitor untuk mengetahui

perbedaan dan persamaan dari aplikasi yang sudah ada dengan yang ingin dikembangkan, serta tujuan yang belum tercapai oleh aplikasi yang sudah ada.

2 Berikut merupakan daftar pertanyaan wawancara kepada mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya untuk Analisis Kebutuhan.

Tabel 3.1 Daftar pertanyaan wawancara untuk Analisis Kebutuhan

No	Pertanyaan
1	Nama
2	Angkatan
3	Apakah anda suka memasak?
4	Makanan apa yang biasa anda masak?
5	Darimana anda belajar memasak makanan tersebut?
6	Apa saja permasalahan yang dihadapi saat memasak?
7	Menurut anda, bagaimana cara anda mengatasi permasalahan tersebut?
8	Apakah menu masakan yang terlalu rumit dapat mempersulit dalam memasak?
9	Apakah jika ada aplikasi yang memberikan resep masakan dan panduan memasaknya, anda akan terbantu dalam mengatasi permasalahan memasak yang mungkin terjadi?
10	Jika resep yang disajikan dalam aplikasi ini sederhana dan tidak terlalu rumit, yang mana lebih spesifik ke mahasiswa, apakah lebih terbantu?
11	Apa saja yang akan anda butuhkan dan ingin lakukan dari aplikasi resep tersebut?
12	Apakah ada kritik dan saran?

2 Pada tabel 3.1 terdapat daftar pertanyaan wawancara kepada mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya untuk Analisis Kebutuhan terkait data diri, dan hal-hal yang berhubungan dengan memasak.

3.7.3 Pengembangan Prototype

Pengembangan *Prototype* dilakukan dengan menggunakan Figma, yang mengacu kepada kebutuhan dan spesifikasi perangkat lunak, yang mana diuraikan pada *persona*, *scenario*, *user story*, dan fitur. Pengembangan *Prototype* bertujuan untuk membangun sebuah desain *prototype* dari aplikasi yang ingin dibuat. *Prototype* tersebut sudah dapat diklik dan digunakan oleh pengguna,

sehingga nantinya dapat dievaluasi kepada pengguna pada tahap evaluasi *prototype*.²

3.7.4 Evaluasi *Prototype*

Pada tahap Evaluasi *Prototype*, *prototype* yang telah dibuat akan diuji kepada pengguna, dan dievaluasi. Nantinya, hasil dari evaluasi akan dijadikan bahan pertimbangan untuk perlu tidaknya dilakukan perbaikan. Jika dibutuhkan perbaikan, maka akan kembali ke tahap Analisis Kebutuhan untuk nantinya akan dibuat kembali *prototype* sesuai dengan perbaikan yang dibutuhkan. Evaluasi *Prototype* bertujuan untuk memvalidasi dan mengetahui apakah *prototype* yang telah dibuat sudah sesuai dengan kebutuhan pengguna.²

Berikut merupakan daftar *task scenario* kepada mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya untuk Evaluasi *Prototype* iterasi pertama.

Tabel 3.2 Daftar *task scenario* untuk Evaluasi *Prototype* iterasi pertama

No	Pertanyaan
1	Coba masuk ke halaman beranda dari halaman <i>login</i>
2	Coba buka resep “nasi goreng spesial” dari halaman beranda
3	Coba kembali ke halaman beranda dari halaman detail resep
4	Coba buka halaman pengaturan dari halaman beranda
5	Coba ubah <i>email</i> dan nama akun di halaman pengaturan
6	Coba ubah <i>password</i> akun di halaman pengaturan
7	Coba keluar akun di halaman pengaturan
8	Apakah ada kritik dan saran terhadap <i>prototype</i> ini?

Pada tabel 3.2 terdapat beberapa *task scenario* yang akan dilakukan oleh mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya untuk tahap Evaluasi *Prototype* iterasi pertama.²

Berikut merupakan daftar *task scenario* kepada mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya untuk Evaluasi *Prototype* iterasi kedua.

Tabel 3.3 Daftar *task scenario* untuk Evaluasi *Prototype* iterasi kedua

No	Pertanyaan
1	Coba buka resep “Nasi Goreng Spesial”
2	Coba masuk ke akun
3	Coba buka halaman lupa <i>password</i> dari <i>login</i> dan kirim permintaan reset

	<i>password</i> ke <i>email</i> kita
4	Coba buka resep “Nasi Goreng Spesial” dan tambahkan catatan pribadi
5	Coba tambahkan <i>rating</i> bintang 5 dan <i>review</i> di resep “Nasi Goreng Spesial”
6	Coba simpan resep “Nasi Goreng Spesial” ke daftar resep tersimpan
7	Coba hapus “Nasi Goreng Spesial” dari daftar resep tersimpan
8	Coba ubah akun, dan ubah <i>password</i> di halaman pengaturan secara berurutan
9	Coba keluar akun di halaman pengaturan
10	Apakah ada kritik dan saran terhadap <i>prototype</i> ini?

Pada tabel 3.3 terdapat beberapa *task scenario* yang akan dilakukan oleh mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya untuk tahap Evaluasi *Prototype* iterasi kedua.

3.7.5 Perancangan

Tahap Perancangan akan merancang perangkat lunak yang akan dikembangkan, berdasarkan *prototype* yang telah tervalidasi pada tahap evaluasi. Pada tahap ini, akan dirancang arsitektur perangkat lunak, *sequence diagram*, *class diagram*, basis data, dan algoritme dalam bentuk *pseudocode*. Perancangan bertujuan untuk membuat sebuah rancangan yang akan dipakai sebagai acuan dalam implementasi pengembangan perangkat lunak pada penelitian ini.

3.7.6 Implementasi

Tahap implementasi akan mengimplementasikan rancangan perangkat lunak menjadi sebuah perangkat lunak. Proses implementasi dilakukan dengan mengembangkan sebuah perangkat lunak berupa aplikasi perangkat bergerak berbasis Android dengan menggunakan Android Studio menggunakan bahasa pemrograman Kotlin. Pada tahap ini, akan dijabarkan spesifikasi sistem yang dipakai dalam lingkungan pengembangan perangkat lunak, implementasi *database*, implementasi kode program, dan implementasi antarmuka pengguna. Implementasi bertujuan untuk menghasilkan sebuah perangkat lunak yang dapat digunakan oleh pengguna untuk mencapai tujuannya.

3.7.7 Pengujian

Perangkat lunak yang telah dibuat selanjutnya akan diuji pada tahap Pengujian. Pengujian bertujuan untuk mengevaluasi kemampuan perangkat lunak, keselarasan perangkat lunak dengan kebutuhan pengguna, dan menentukan kesesuaian dengan hasil yang diharapkan. Pengujian pada

penelitian ini terdiri dari dua macam pengujian, yaitu *black box testing*, dan *usability testing*. *Black box testing* adalah sebuah pendekatan pengujian dimana pengujian dilakukan dengan membuat kasus uji terhadap setiap fungsi, guna memastikan keselarasan semua alur perangkat lunak dengan kebutuhan pengguna. *Black box testing* dilakukan dengan menggunakan metode *scenario-based testing* yang akan diuji kepada pengembang. *Usability testing* adalah sebuah pengujian yang digunakan untuk mengetahui apakah sebuah sistem telah memenuhi kebutuhan pengguna, menguji apakah pengguna dapat mengerti cara menggunakan sistem secara cepat tanpa kesalahan, mendapatkan umpan balik secara cepat dalam meningkatkan antarmuka, dan melakukan koreksi terhadap kesalahan dalam komponen perangkat. *Usability testing* dilakukan menggunakan *scenario-based testing* secara *moderated usability testing* yang akan diuji kepada pengguna. Metode pengukuran *usability testing* yang digunakan pada penelitian ini yaitu efektivitas dan *System Usability Scale*. *Scenario-based testing* adalah sebuah pengujian yang berbasis skenario yang berfokus pada pengujian alur bisnis perangkat lunak. Skenario pada *scenario-based testing* menyerupai instruksi terkait bagaimana perangkat lunak digunakan.

2

3.7.8 Kesimpulan dan Saran

6 Tahap kesimpulan dan saran dilakukan dengan menyimpulkan tahapan-tahapan yang telah dilakukan dan hasil dari penelitian ini. Kesimpulan dan saran mengacu dari hasil yang didapat pada tahap pengujian. Kesimpulan bertujuan untuk mengetahui inti dan rangkuman informasi dari penelitian ini. Saran bertujuan untuk memberi masukan terhadap hasil dari penelitian ini, sehingga diharapkan apabila diterapkan akan menghasilkan penelitian yang lebih baik.

2 BAB 4 ANALISIS KEBUTUHAN

4.1 Analisis Kompetitor

Analisis kompetitor dilakukan untuk mengetahui perbedaan dan persamaan dari aplikasi yang sudah ada dengan yang ingin dikembangkan, serta tujuan yang belum tercapai oleh aplikasi yang sudah ada. Hal ini dimaksudkan untuk membandingkan aplikasi tersebut, dan mencari peluang keunikan yang tidak dimiliki oleh kompetitor. Pada penelitian ini dilakukan analisis kompetitor terhadap salah satu aplikasi resep yaitu Cookpad.

Tabel 4.1 Analisis kompetitor terhadap aplikasi Cookpad

Nama Aplikasi	Cookpad
Persamaan	<ul style="list-style-type: none">- Terdapat fitur pencarian- Terdapat fitur simpan resep- Terdapat gambar, nama, bahan-bahan, langkah pembuatan, dan estimasi waktu pembuatan resep- Terdapat fitur komentar terhadap resep- Fitur melihat daftar dan detail resep, dan pencarian resep untuk pengguna yang belum masuk akun- Fitur penilaian dan komentar terhadap resep, dan simpan resep terbatas pada pengguna yang sudah masuk akun
Perbedaan	<ul style="list-style-type: none">- Penilaian terhadap resep menggunakan emoji, sedangkan aplikasi pada penelitian ini menggunakan <i>rating</i> dengan skala 1-5- Resep yang disajikan umum, sedangkan aplikasi pada penelitian ini resep yang disajikan dikhkususkan untuk mahasiswa, dimana resep tersebut lebih sederhana dan tidak rumit- Fitur simpan resep terbatas pada 60 resep jika tidak berlangganan berbayar, sedangkan pada aplikasi ini tidak ada batasan menyimpan resep
Tujuan yang belum tercapai	<ul style="list-style-type: none">- Daftar harga tiap bahan- Estimasi total harga bahan- Rekomendasi berdasarkan estimasi total harga bahan- Catatan pribadi tiap resep- Semua fitur dapat diakses secara gratis, termasuk simpan resep tak terbatas

Pada tabel 4.1 terdapat analisis kompetitor terhadap aplikasi Cookpad, dimana didapatkan beberapa persamaan, perbedaan, dan tujuan yang belum tercapai dibandingkan aplikasi yang ingin dikembangkan pada penelitian ini.

4.2 Pengembangan *Prototype* Iterasi Pertama

Dalam pengembangan *prototype* iterasi pertama, digunakan analisis kebutuhan yang mengacu dari wawancara kepada pengguna. Hasil dari wawancara dapat dilihat pada Lampiran B.1. Pada tahap iterasi pertama dirancang *persona*, *scenario*, *user story*, fitur, dan *prototype* aplikasi dalam bentuk *high-fidelity*.

4.2.1 *Persona*

Persona adalah sebuah ‘pengguna khayalan’, potret karakter dari tipe pengguna yang mungkin akan menggunakan sebuah, yang mana akan membantu membayangkan apa saja yang mungkin pengguna inginkan dari perangkat lunak, bagaimana mereka menggunakaninya, dan kesulitan yang mungkin akan dialami dalam memahami dan menggunakan fitur. Pada penelitian ini terdapat satu *persona* yaitu Miko, seorang mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya atau dapat disingkat FILKOM UB.

Tabel 4.2 *Persona* pada *prototype* iterasi pertama

Miko, seorang mahasiswa FILKOM UB
Miko, 21 tahun, adalah seorang mahasiswa teknik informatika FILKOM UB angkatan 2019. Ia juga merupakan seorang anak kos di Malang. Semenjak kuliah dan ngekos, ia jadi suka memasak demi menghemat biaya. Beberapa makanan yang biasa ia masak antara lain mie goreng, nasi goreng, ayam teriyaki, bakso, dan terkadang spageti. Dalam memasak, dia menggunakan <i>smartphone</i> -nya untuk mencari resep di Google, dan video <i>tutorial</i> di Youtube. Namun, terkadang masakan yang ia buat tidak berhasil sesuai ekspektasi. Penyebabnya antara lain bingung soal langkah-langkah memasak, istilah yang asing di resep, dan sulit untuk mengetahui takaran bumbu yang benar. Seringkali ia juga kesulitan saat memasak makanan yang terlalu rumit karena menu makanan yang rumit memiliki bumbu yang rumit, dengan cara masak yang rumit pula. Ia mungkin berpikir untuk menggunakan sebuah aplikasi penyedia resep masakan sederhana, yang dapat memandunya memasak masakan yang cocok untuk mahasiswa, dimana menu yang tersedia tidak terlalu rumit baginya. Ia juga ingin mengetahui rincian harga tiap bahan dan waktu estimasi yang dibutuhkan dalam memasak suatu resep.

Pada tabel 4.2 terdapat sebuah *persona* yang bernama Miko yang merupakan seorang mahasiswa FILKOM UB, yang termasuk di dalamnya narasi terkait aspek

personalisasi, relevansi, edukasi, dan pekerjaan. Aspek personalisasi direpresentasikan dengan identitasnya sebagai seseorang mahasiswa kos yang berumur 21 tahun. Aspek relevansi direpresentasikan dengan hobinya dalam memasak dan menggunakan *smartphone* dalam mencari resep masakan, dan beberapa permasalahan yang ia alami terkait hal tersebut, beserta keinginan untuk menyelesaiakannya. Aspek edukasi dan pekerjaan direpresentasikan dengan narasi bahwa ia merupakan mahasiswa teknik informatika FILKOM UB angkatan 2019.

4.2.2 Scenario

Scenario adalah sebuah narasi mengenai situasi yang mana pengguna menggunakan fitur dari sebuah produk untuk melakukan hal yang mereka inginkan, yang mana memaparkan masalah pengguna, dan menunjukkan cara yang dibayangkan dalam penyelesaian masalah tersebut. Berikut merupakan sebuah *scenario* yang menggambarkan kegiatan memasak Miko.

Tabel 4.3 *Scenario* pada *prototype* iterasi pertama

Memasak nasi goreng
Miko adalah seorang mahasiswa teknik informatika FILKOM UB angkatan 2019 yang ngekos di Malang. Ia mulai belajar memasak semenjak masuk kuliah dan ngekos.
Ia menggunakan resep dan tutorial yang ia temukan di internet untuk belajar memasak. Namun, seringkali ia bingung ingin memasak apa. Ia memilih resep masakan yang ingin ia masak berdasarkan rekomendasi di internet.
Suatu malam, saat sedang akhir bulan dan kehabisan uang, ia hanya memiliki persediaan nasi, daging mentah, dan bumbu dapur untuk makan. Ia mencari rekomendasi resep masakan di internet dan memilih nasi goreng untuk menu masakannya malam tersebut. Namun, setiap kali ia memasak nasi goreng, hasilnya tidak pernah memuaskan. Entah karena rasanya terlalu hambar, terlalu asin, maupun nasi yang terlalu keras. Seringkali resep yang ia pakai membutuhkan bumbu yang terlalu rumit sehingga ia kesulitan dalam mengolahnya. Permasalahan lain yang ia hadapi adalah masalah harga dari bahan-bahan pada suatu resep. Ia harus menyesuaikan bahan yang terjangkau untuknya demi penghematan.
Untuk menyelesaikan permasalahan-permasalahan tersebut, ia memilih untuk mencari sebuah aplikasi resep yang sederhana dengan takaran yang jelas, disertai harga dari tiap bahan yang dibutuhkan dari resep tersebut. Ia bertanya kepada temannya Ziyad apakah ada aplikasi yang memenuhi keinginannya tersebut. Ziyad merekomendasikan sebuah aplikasi resep yang dikhususkan untuk mahasiswa, dimana resep yang disajikan sederhana, disertai dengan

rincian harga dari bahannya, dan cocok untuk mahasiswa yang ngekos. Miko menggunakan aplikasi resep dan panduan masakan untuk mahasiswa tersebut dan menggunakan resep di sana untuk memasak nasi goreng dengan varian yang sesuai dengan ketersediaan bahan yang ia punya.

Pada tabel 4.3 terdapat sebuah *scenario* yang menggambarkan kegiatan Miko dalam hal memasak nasi goreng, yang termasuk di dalamnya nama skenario, tujuan keseluruhan, hal yang terlibat dalam mencapai tujuan, persona dari aktor yang terlibat, permasalahan yang tidak bisa terselesaikan oleh sistem yang sudah ada, dan cara yang memungkinkan dalam menyelesaikan permasalahan tersebut. Nama skenarionya adalah memasak nasi goreng. Tujuan keseluruhannya adalah memasak nasi goreng sederhana yang diinginkan dengan ketersediaan bahan dan harganya. Hal yang terlibat dalam mencapai tujuan antara lain pencarinya dalam mencari resep di internet, dan bertanya mengenai aplikasi resep kepada temannya Ziyad. Persona dari aktor yang terlibat direpresentasikan dengan paragraf awal mengenai Miko. Permasalahan dari yang tidak bisa terselesaikan oleh sistem yang sudah ada yaitu mengenai belum adanya aplikasi resep yang dikhususkan untuk mahasiswa beserta rekomendasi resep sesuai estimasi harga bahannya, bahan masakan dalam resep yang terlalu rumit, dan harga dari tiap bahan pada resep yang tidak dicantumkan. Cara yang memungkinkan dalam menyelesaikan permasalahan yaitu dengan adanya sebuah aplikasi resep untuk mahasiswa, dimana resep yang disajikan sederhana, disertai dengan rincian harga dari bahannya, dan cocok untuk mahasiswa yang ngekos.

4.2.3 User Story

User story adalah sebuah deskripsi mengenai situasi dimana pengguna mencoba melakukan sesuatu dengan sistem perangkat lunak. Berikut merupakan *user story* yang dikembangkan dari *scenario* memasak nasi goreng.

Tabel 4.4 User story pada prototype iterasi pertama

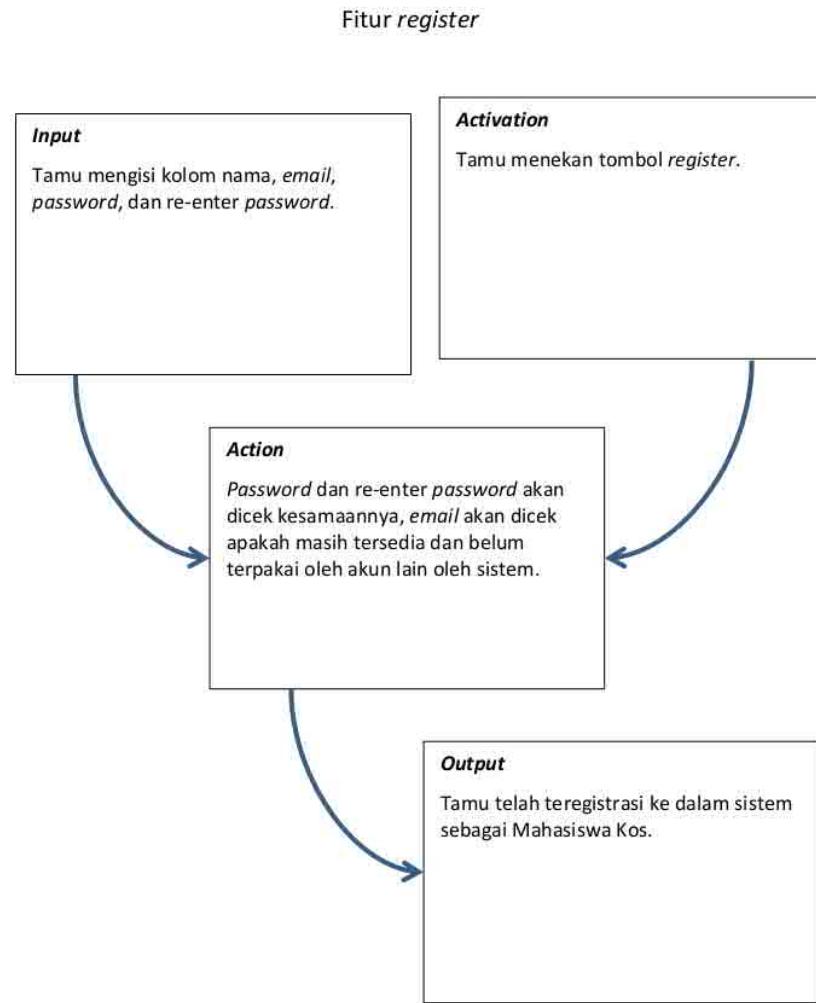
No	User Story
1.	Sebagai pengguna baru, saya ingin dapat melakukan pendaftaran pada sistem dengan membuat nama, <i>email</i> , dan <i>password</i> sehingga sistem dapat menyimpan data akun saya.
2.	Sebagai pengguna yang telah terdaftar, saya ingin <i>masuk ke dalam sistem dengan memasukkan email, dan password</i> sehingga sistem dapat mengautentikasi saya.
3.	Sebagai pengguna yang telah terdaftar, saya ingin mencari resep masakan yang sesuai untuk mahasiswa, dimana resep tersebut sederhana dan tidak rumit, dengan takaran yang jelas.

4.	Sebagai pengguna yang telah terdaftar, saya ingin mendapatkan rekomendasi resep masakan berdasarkan harga.
5.	Sebagai pengguna yang telah terdaftar, saya ingin mengetahui bahan-bahan apa saja yang dibutuhkan dari suatu resep, harga dari tiap bahan, beserta langkah-langkah memasaknya.

Pada tabel 4.4 terdapat beberapa *user story* yang dikembangkan dari *scenario* memasak nasi goreng. Beberapa *user story* tersebut merupakan *refined user story* atau *user story* yang sudah disempurnakan. Tiap *user story* menggambarkan detail dari penggunaan fitur pada aplikasi.

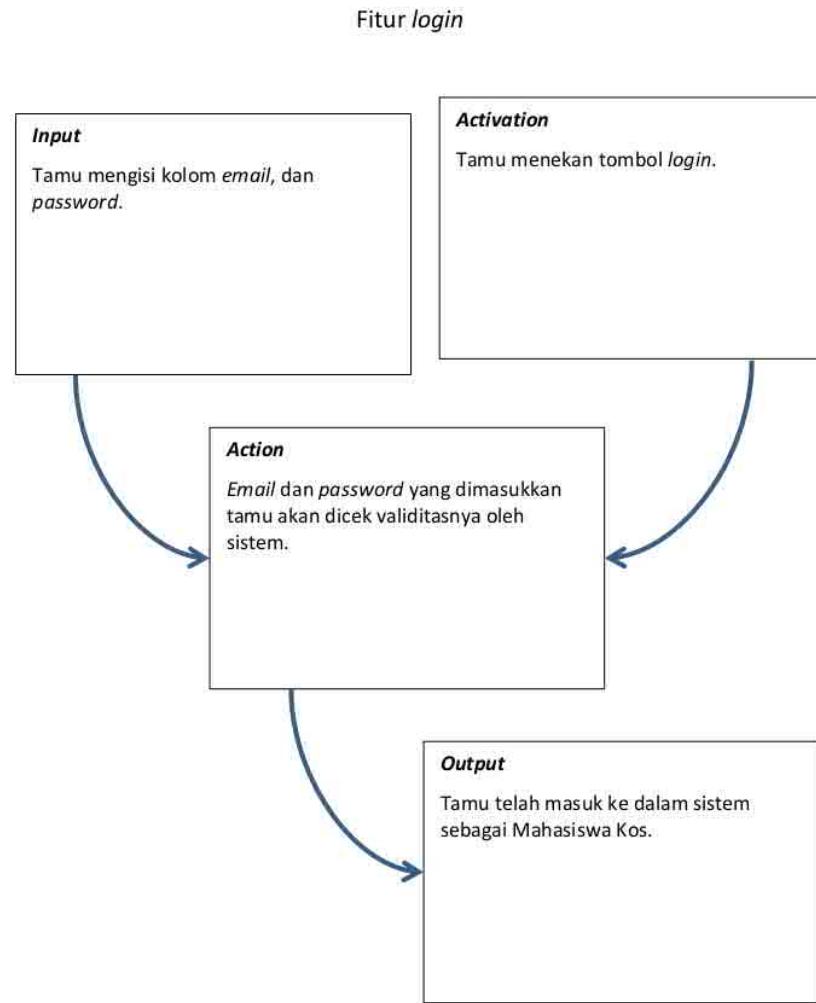
4.2.4 Fitur

Fitur adalah sebuah cara yang memungkinkan pengguna mengakses dan menggunakan fungsionalitas produk, sehingga daftar fitur mendefinisikan fungsionalitas sistem secara keseluruhan. Berikut merupakan fitur-fitur yang mengacu pada *user story*.



Gambar 4.1 Fitur *register* pada *prototype* iterasi pertama

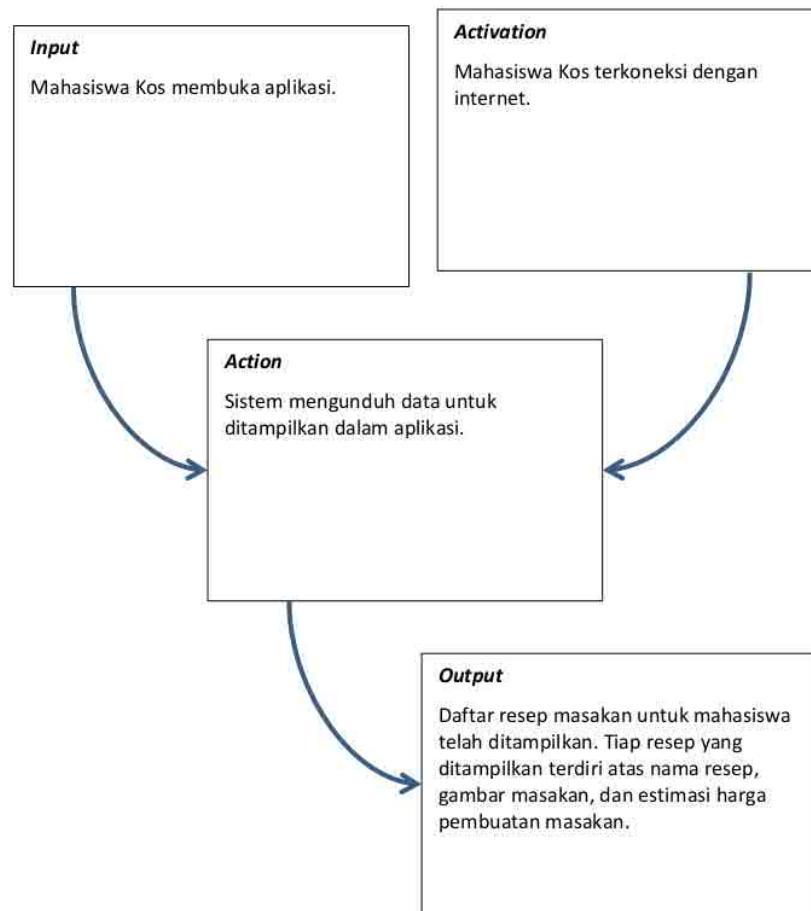
Pada gambar 4.1 terdapat fitur *register* yang mengacu pada *user story*.



Gambar 4.2 Fitur *login* pada prototype iterasi pertama

Pada gambar 4.2 terdapat fitur *login* yang mengacu pada *user story*.

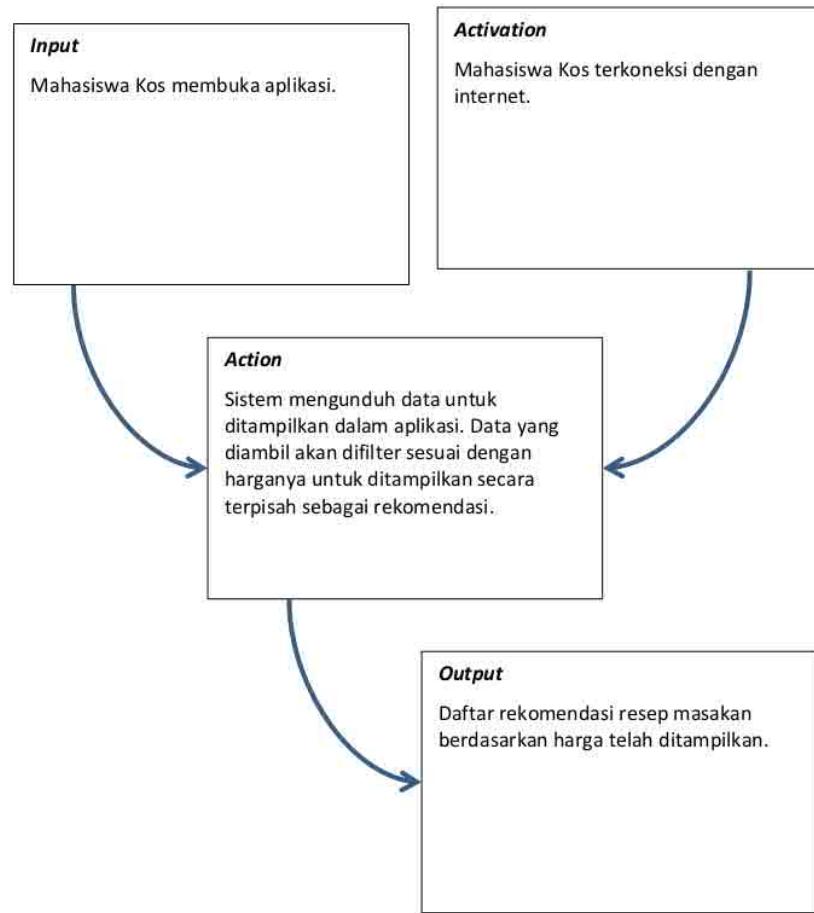
Fitur menampilkan daftar resep masakan untuk mahasiswa



Gambar 4.3 Fitur menampilkan daftar resep masakan untuk mahasiswa pada prototype iterasi pertama

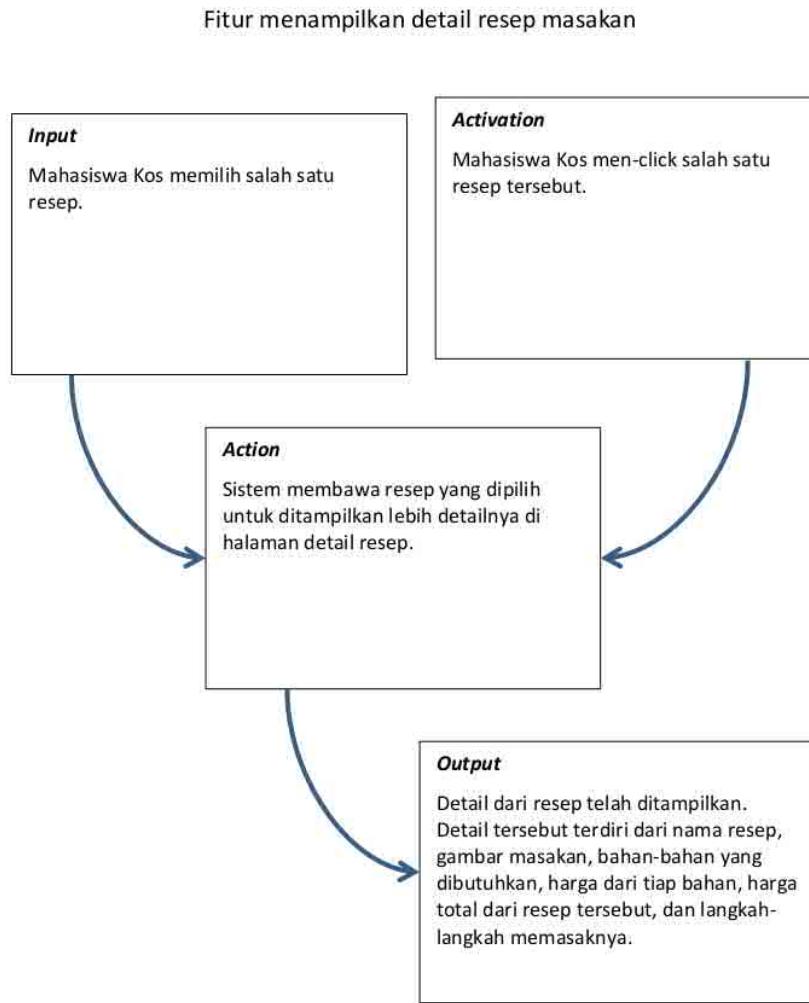
Pada gambar 4.3 terdapat fitur menampilkan daftar resep masakan untuk mahasiswa yang mengacu pada *user story*.

Fitur menampilkan rekomendasi resep masakan berdasarkan harga



Gambar 4.4 Fitur menampilkan rekomendasi resep masakan berdasarkan harga pada *prototype* iterasi pertama

Pada gambar 4.4 terdapat fitur menampilkan rekomendasi resep masakan berdasarkan harga yang mengacu pada *user story*.



Gambar 4.5 Fitur menampilkan detail resep masakan pada *prototype* iterasi pertama

Pada gambar 4.5 terdapat fitur menampilkan detail resep masakan berdasarkan harga yang mengacu pada *user story*.

4.2.5 Prototype

Pada subbab ini akan dijabarkan sebuah desain *prototype* dari aplikasi yang mengacu kepada *persona*, *scenario*, *user story*, dan fitur. *Prototype* tersebut sudah dapat diklik dan digunakan oleh pengguna, sehingga nantinya dapat dievaluasi kepada pengguna pada tahap evaluasi *prototype* iterasi pertama.

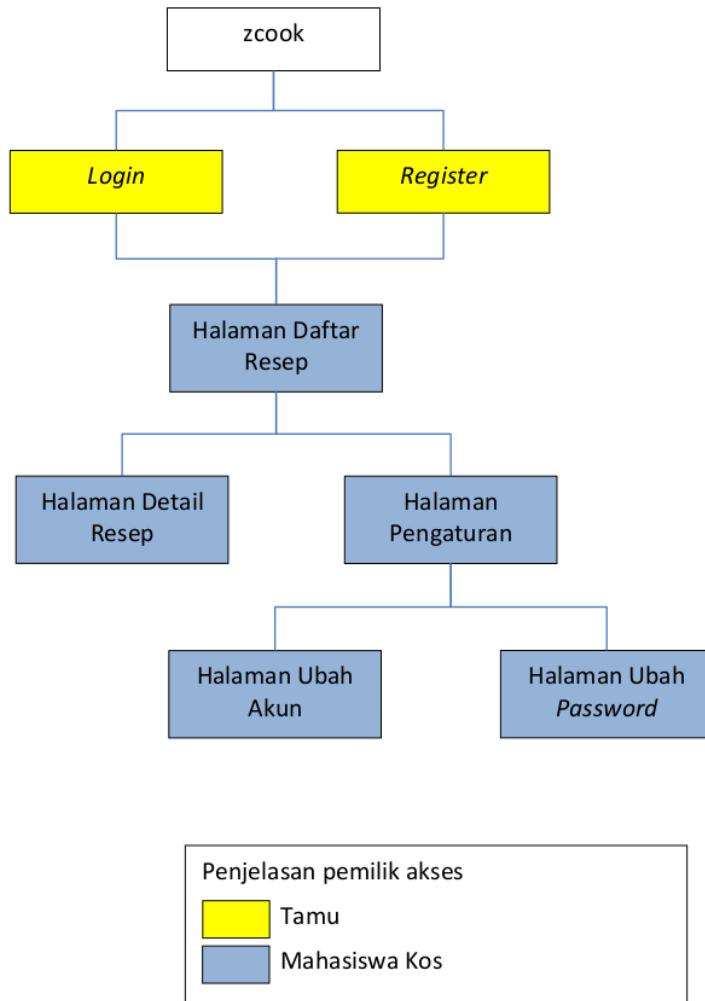
Pada aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa ini terdapat pengguna yang akan menjalankan sistem. Pengguna yang akan menggunakan sistem ini diuraikan pada tabel 4.5.

Tabel 4.5 Identifikasi pengguna pada prototype iterasi pertama

Jenis Pengguna	Deskripsi
Tamu	Tamu merupakan pengguna yang belum masuk ke dalam sistem. Tamu dapat melakukan registrasi dan login agar dapat menggunakan fitur pada aplikasi.
Mahasiswa Kos	Mahasiswa Kos merupakan pengguna yang telah terdaftar dalam sistem. Pengguna ini dapat melihat resep dari daftar resep yang tersedia.

Pada tabel 4.5 terdapat beberapa jenis pengguna yang akan menggunakan sistem, yang mana terdiri dari Tamu dan Mahasiswa Kos.

Berikut merupakan *information architecture* sistem.

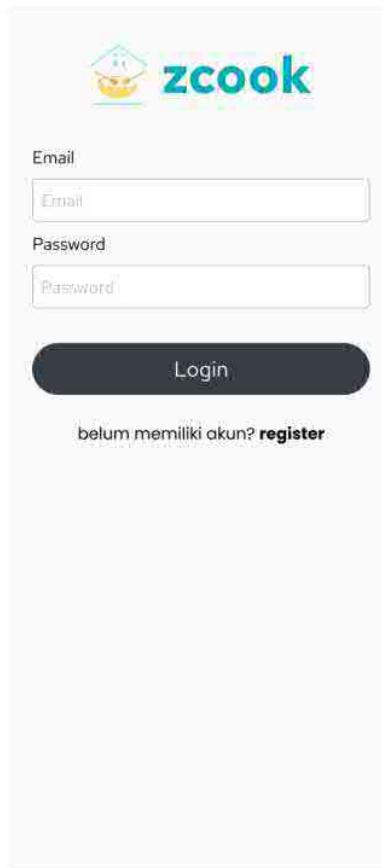


2 **Gambar 4.6 Information architecture sistem pada prototype iterasi pertama**

2 Pada gambar 4.6 terdapat information architecture sistem yang terdiri dari beberapa halaman, yang dikategorikan sesuai pemilik aksesnya.

65 **4.2.5.1 Desain Halaman Login**

Berikut merupakan desain halaman *login*.

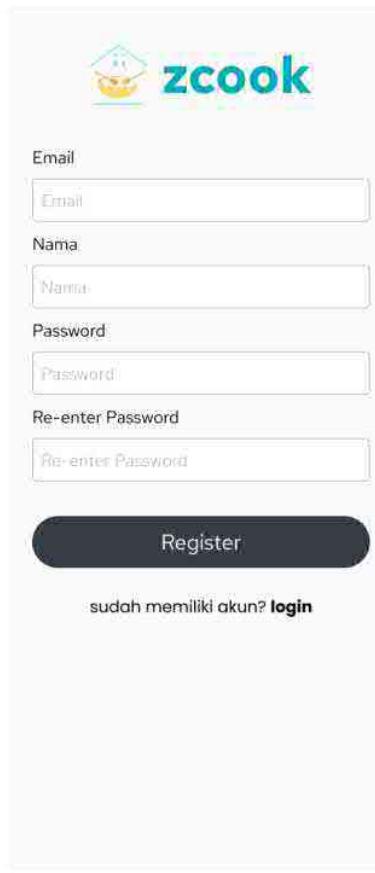


Gambar 4.7 Desain halaman *login* pada *prototype* iterasi pertama

Pada halaman *login*, terdapat kolom untuk mengisi *email* dan *password* yang nantinya akan diisi oleh pengguna, dan divalidasi oleh sistem saat tombol *login* ditekan. Terdapat pula teks *register* yang menavigasikan pengguna ke halaman *register*.

4.2.5.2 Desain Halaman *Register*

Berikut merupakan desain halaman *register*.

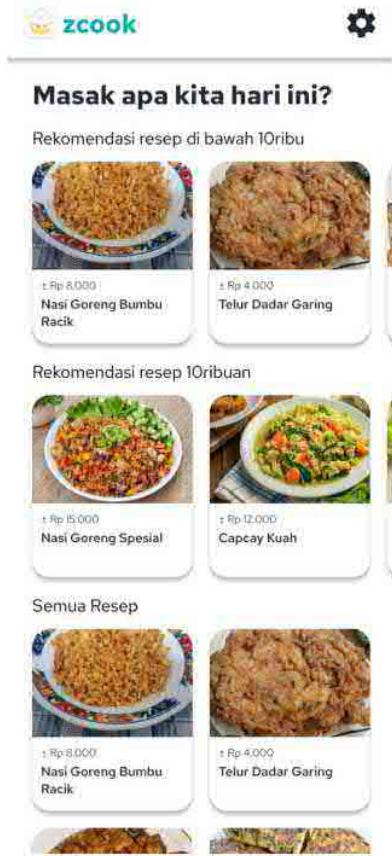


Gambar 4.8 Desain halaman **register** pada **prototype iterasi pertama**

Pada halaman **register**, terdapat kolom untuk mengisi **email**, **nama**, **password**, dan **re-enter password** yang nantinya akan diisi oleh pengguna, dan divalidasi oleh sistem saat tombol **register** ditekan. **Password** dan **re-enter password** akan dicek kesamaannya, dan **email** akan dicek apakah masih tersedia dan belum terpakai oleh akun lain oleh sistem. Terdapat pula teks **login** yang menavigasikan pengguna ke halaman **login**.

4.2.5.3 Desain Halaman Daftar Resep

Berikut merupakan desain halaman daftar resep.



Gambar 4.9 Desain halaman daftar resep pada *prototype* iterasi pertama

Pada halaman daftar resep, terdapat rekomendasi resep sesuai harganya, dalam hal ini resep 10ribuan dan resep di bawah 10ribu, dan juga terdapat daftar semua resep yang tersedia. Tiap resep yang ditampilkan terdiri atas nama resep, gambar masakan, dan estimasi harga pembuatan masakan. Resep yang ditekan akan navigasikan pengguna ke halaman detail dari resep tersebut. Terdapat juga tombol pengaturan yang akan navigasikan pengguna ke halaman pengaturan.

4.2.5.4 Desain Halaman Detail Resep

Berikut merupakan desain halaman *login*.



Nasi Goreng

Total harga: ± Rp 15.000

Bahan:

- Nasi putih 1 piring (± Rp 3.000)
- Bawang putih 2 siung, cincang halus (± Rp 500)
- Kecap manis 1 sdm atau sesuai selera (± Rp 1000 persachet 20ml)
- Saus sambal 1 sdm atau sesuai selera (± Rp 500 persachet 10gr)
- Saus tiram 1 sdm atau sesuai selera (± Rp 3.000 persachet 23ml)
- Garam 1 sdt atau secukupnya (± Rp 2.000 persachet 200gr)
- Kaldu bubuk rasa ayam atau sapi 1 sdt atau sesuai selera (± Rp 500 persachet 8gr)
- Daun bawang 1 batang, cincang halus (± Rp 1.000 per 80gr)
- Telur ayam 1 butir (± Rp 2.000)
- Sosis ayam 1 buah, iris tipis (± Rp 1.000)
- Margarin atau minyak goreng 3 sdm (± Rp 500)

Cara membuat:

1. Siapkan pengorengan dengan api sedang, tuang margarin atau minyak goreng.

Gambar 4.10 Desain halaman detail resep pada prototype iterasi pertama

Pada halaman detail resep, terdapat nama resep, gambar masakan, bahan-bahan yang dibutuhkan, harga dari tiap bahan, harga total dari resep tersebut, dan langkah-langkah memasaknya. Terdapat juga tombol kembali, untuk kembali ke halaman sebelumnya.

4.2.5.5 Desain Halaman Pengaturan

Berikut merupakan desain halaman pengaturan.



Gambar 4.11 Desain halaman pengaturan pada *prototype* iterasi pertama

Pada halaman pengaturan, terdapat nama pengguna, *email* pengguna, tombol ubah akun, tombol ubah *password*, dan tombol keluar akun. Terdapat juga tombol kembali, untuk kembali ke halaman sebelumnya. Tombol ubah akun akan navigasikan pengguna ke halaman ubah akun, tombol ubah *password* akan navigasikan pengguna ke halaman ubah *password*, dan tombol keluar akun akan mengeluarkan pengguna dari hak akses pengguna Mahasiswa Kos dan navigasikan pengguna ke halaman *login*.

4.2.5.6 Desain Halaman Ubah Akun

Berikut merupakan desain halaman ubah akun.

Ubah Akun

Nama
Muhammad Ziyad Al Maududi

Email
mziyad195@gmail.com

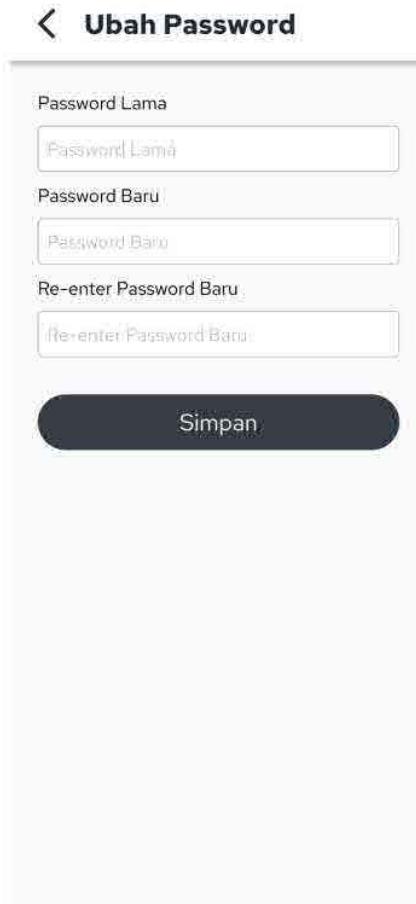
Simpan

Gambar 4.12 Desain halaman ubah akun pada *prototype* iterasi pertama

Pada halaman ubah akun, terdapat kolom untuk mengisi nama, dan email, yang nantinya akan diisi oleh pengguna, dan diproses untuk disimpan oleh sistem saat tombol simpan ditekan. Terdapat juga tombol kembali, untuk kembali ke halaman sebelumnya.

4.2.5.7 Desain Halaman Ubah Password

Berikut merupakan desain halaman ubah password.



Gambar 4.13 Desain halaman ubah password pada prototype iterasi pertama

Pada halaman ubah password, terdapat kolom untuk mengisi *password lama*, *password baru*, dan *re-enter password baru*, yang nantinya akan diisi oleh pengguna, dan diproses untuk disimpan oleh sistem saat tombol simpan ditekan. *Password lama* akan dicek oleh sistem validitasnya, dan *password* dan *re-enter password* akan dicek kesamaannya. Terdapat juga tombol kembali, untuk kembali ke halaman sebelumnya.

4.2.6 Evaluasi Prototype

Setelah *prototype* yang telah dibuat diuji kepada pengguna, didapatkan beberapa evaluasi. Hasil dari evaluasi akan dijadikan acuan perbaikan pada pengembangan *prototype* iterasi kedua. Beberapa perbaikan yang dilakukan adalah perbaikan dari sisi akses pengguna pada *information architecture*, perubahan *scenario*, penambahan *user story* beserta fitur non-fungsional terkait versi android, dan penggunaan *refined user story*. Selain itu, terdapat pula

penambahan fitur pencarian resep, resep favorit, *rating* dan *review*, catatan pribadi dari resep, dan perbaikan pada desain *prototype*. Hasil dari evaluasi *prototype* iterasi pertama dapat dilihat pada Lampiran C.1.

4.3 Pengembangan *Prototype* Iterasi Kedua

Dalam pengembangan *prototype* iterasi kedua, dilakukan kembali tahap analisis kebutuhan yang mengacu dari evaluasi *prototype* iterasi pertama. Pada tahap iterasi kedua dirancang *scenario*, *user story*, fitur, dan *prototype* aplikasi dalam bentuk *high-fidelity*. Tidak ada perubahan dari sisi persona sehingga tidak dituliskan kembali di sini.

4.3.1 *Scenario*

Scenario adalah sebuah narasi mengenai situasi yang mana pengguna menggunakan fitur dari sebuah produk untuk melakukan hal yang mereka inginkan, yang mana memaparkan masalah pengguna, dan menunjukkan cara yang dibayangkan dalam penyelesaian masalah tersebut. Berikut merupakan sebuah *scenario* yang menggambarkan kegiatan memasak Miko.

Tabel 4.6 *Scenario* pada *prototype* iterasi kedua

Memasak nasi goreng
Miko adalah seorang mahasiswa teknik informatika FILKOM UB angkatan 2019 yang ngekos di Malang. Ia mulai belajar memasak semenjak masuk kuliah dan ngekos.
Ia menggunakan resep dan tutorial yang ia temukan di internet untuk belajar memasak. Namun, seringkali ia bingung ingin memasak apa. Ia memilih resep masakan yang ingin ia masak berdasarkan rekomendasi di internet.
Suatu malam, saat sedang akhir bulan dan kehabisan uang, ia hanya memiliki persediaan nasi, daging mentah, dan bumbu dapur untuk makan. Ia mencari rekomendasi resep masakan di internet dan memilih nasi goreng untuk menu masakannya malam tersebut. Namun, setiap kali ia memasak nasi goreng, hasilnya tidak pernah memuaskan. Entah karena rasanya terlalu hambar, terlalu asin, maupun nasi yang terlalu keras. Seringkali resep yang ia pakai membutuhkan bumbu yang terlalu rumit sehingga ia kesulitan dalam mengolahnya. Permasalahan lain yang ia hadapi adalah masalah harga dari bahan-bahan pada suatu resep. Ia harus menyesuaikan bahan yang terjangkau untuknya demi penghematan.
Untuk menyelesaikan permasalahan-permasalahan tersebut, ia memilih untuk mencari sebuah aplikasi resep yang sederhana dengan takaran yang jelas, disertai harga dari tiap bahan yang dibutuhkan dari resep tersebut. Ia juga memilih untuk mencari aplikasi resep yang dapat mencari menu yang diinginkan,

menyimpan menu favorit, *rating* dan *review* dari tiap menu sehingga ia mengetahui kualitas sebuah resep, estimasi waktu yang dibutuhkan saat memasak, dan catatan pribadi dari sebuah resep sehingga ia dapat mencatat pendapatnya tentang suatu resep untuk dirinya sendiri. Ia juga menginginkan aplikasi resep yang menyediakan fitur terkait manajemen akun seperti mengubah nama pengguna, *email* pengguna, *password*, dan fitur lupa *password* jika sewaktu-waktu ia lupa akan *passwordnya*. Ia bertanya kepada temannya Ziyad apakah ada aplikasi yang memenuhi keinginannya tersebut. Tentunya aplikasi yang ia cari adalah yang *compatible* dengan *handphone* miliknya yaitu Asus Zenfone 2 Laser dengan versi Android 5 Lollipop dengan versi API 21. Ziyad merekomendasikan sebuah aplikasi resep yang dikhususkan untuk mahasiswa, dimana resep yang disajikan sederhana, disertai dengan rincian harga dari bahannya, dan cocok untuk mahasiswa yang ngekos. Miko menggunakan aplikasi resep dan panduan masakan untuk mahasiswa tersebut dan menggunakan resep di sana untuk memasak nasi goreng dengan varian yang sesuai dengan ketersediaan bahan yang ia punya.

Pada tabel 4.6 terdapat sebuah *scenario* yang menggambarkan kegiatan Miko dalam hal memasak nasi goreng, yang termasuk di dalamnya nama skenario, tujuan keseluruhan, hal yang terlibat dalam mencapai tujuan, persona dari aktor yang terlibat, permasalahan yang tidak bisa terselesaikan oleh sistem yang sudah ada, dan cara yang memungkinkan dalam menyelesaikan permasalahan tersebut. Nama skenarionya adalah memasak nasi goreng. Tujuan keseluruhannya adalah memasak nasi goreng sederhana yang diinginkan dengan ketersediaan bahan dan harganya, di sisi lain juga dapat mencari menu yang diinginkan, menyimpan menu favorit, *rating* dan *review* dari tiap menu, estimasi waktu yang dibutuhkan saat memasak, dan catatan pribadi dari sebuah resep. Hal yang terlibat dalam mencapai tujuan antara lain pencarinya dalam mencari resep di internet, dan bertanya mengenai aplikasi resep kepada temannya Ziyad. Persona dari aktor yang terlibat direpresentasikan dengan paragraf awal mengenai Miko. Permasalahan dari yang tidak bisa terselesaikan oleh sistem yang sudah ada yaitu mengenai belum adanya aplikasi resep yang dikhususkan untuk mahasiswa beserta rekomendasi resep sesuai estimasi harga bahannya, bahan masakan dalam resep yang terlalu rumit, dan harga dari tiap bahan pada resep yang tidak dicantumkan. Cara yang memungkinkan dalam menyelesaikan permasalahan yaitu dengan adanya sebuah aplikasi resep untuk mahasiswa, dimana resep yang disajikan sederhana, disertai dengan rincian harga dari bahannya, dan cocok untuk mahasiswa yang ngekos. Aplikasi resep tersebut juga disertai dengan fitur rekomendasi resep berdasarkan estimasi harga bahan, pencarian resep, simpan resep, catatan pribadi, serta *rating* dan *review*.

4.3.2 User Story

User Story adalah sebuah deskripsi mengenai situasi dimana pengguna mencoba melakukan sesuatu dengan sistem perangkat lunak. Berikut merupakan *user story* yang dikembangkan dari *scenario* memasak nasi goreng.

Tabel 4.7 User story pada prototype iterasi kedua

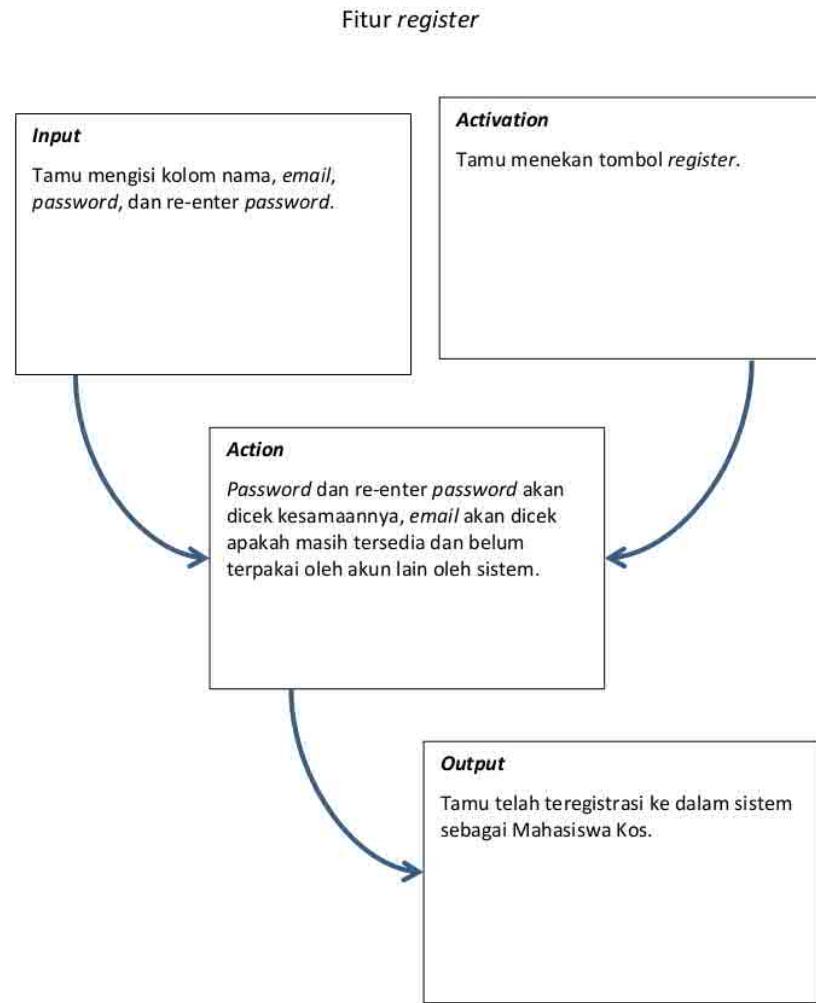
No	User Story
1.	Sebagai pengguna baru, saya ingin dapat melakukan pendaftaran pada sistem dengan membuat nama, <i>email</i> , dan <i>password</i> sehingga sistem dapat menyimpan data akun saya.
2.	Sebagai pengguna yang telah terdaftar, saya ingin masuk ke dalam sistem dengan memasukkan <i>email</i> , dan <i>password</i> sehingga sistem dapat mengautentikasi saya.
3.	Sebagai pengguna yang telah terdaftar, saya ingin dapat mengganti <i>password</i> akun sehingga menjaga akun saya tetap aman.
4.	Sebagai pengguna yang telah terdaftar, saya ingin dapat mengganti nama dan <i>email</i> akun sehingga informasi akun saya dapat diperbarui.
5.	Sebagai pengguna yang telah terdaftar, saya ingin dapat meminta <i>password</i> baru sehingga saya tidak akan kehilangan akses ke akun saya secara permanen jika saya lupa.
6.	Sebagai pengguna, saya ingin dapat melihat daftar resep masakan yang sesuai untuk mahasiswa, dimana resep tersebut sederhana dan tidak rumit, disertai <i>rating</i> , dengan takaran yang jelas sehingga saya dapat menemukan resep yang saya inginkan.
7.	Sebagai pengguna, saya ingin dapat mencari resep masakan sehingga saya dapat menemukan resep yang saya cari.
8.	Sebagai pengguna, saya ingin mendapatkan rekomendasi resep masakan berdasarkan harga sehingga saya mendapatkan rekomendasi dalam menentukan resep yang ingin saya masak.
9.	Sebagai pengguna, saya ingin mengetahui bahan-bahan apa saja yang dibutuhkan dari suatu resep, harga dari tiap bahan, langkah-langkah memasaknya, estimasi waktu yang dibutuhkan, beserta <i>rating</i> dan <i>reviewnya</i> , sehingga saya mengetahui resep memasak secara lengkap.
10.	Sebagai pengguna yang telah masuk, saya ingin dapat menambahkan <i>rating</i> dan <i>review</i> terhadap sebuah resep sehingga pengguna lain dapat mengetahui penilaian saya terhadap resep tersebut.

11.	Sebagai pengguna yang telah masuk, saya ingin dapat menyimpan resep sehingga saya dapat menemukannya secara mudah apabila ingin melihatnya lagi.
12.	Sebagai pengguna yang telah masuk, saya ingin dapat menghapus resep dari resep yang tersimpan sehingga saya dapat menghilangkannya dari daftar resep tersimpan.
13.	Sebagai pengguna yang telah masuk, saya ingin dapat menambahkan catatan pribadi dari tiap resep sehingga saya dapat mencatat pendapatnya tentang suatu resep untuk dirinya sendiri.
14.	Sebagai pengguna yang telah masuk, saya ingin dapat keluar dari akun saya.
15.	Sebagai pengguna, saya ingin dapat menggunakan aplikasi ini pada minimum versi Android 5 Lollipop dengan versi API 21 sehingga saya sebagai pengguna <i>handphone</i> dengan versi Android 5 Lollipop dapat menggunakannya.

Pada tabel 4.7 terdapat beberapa *user story* yang dikembangkan dari *scenario* memasak nasi goreng. Beberapa *user story* tersebut merupakan *refined user story* atau *user story* yang sudah disempurnakan. Tiap *user story* menggambarkan detail dari penggunaan fitur pada aplikasi.

4.3.3 Fitur

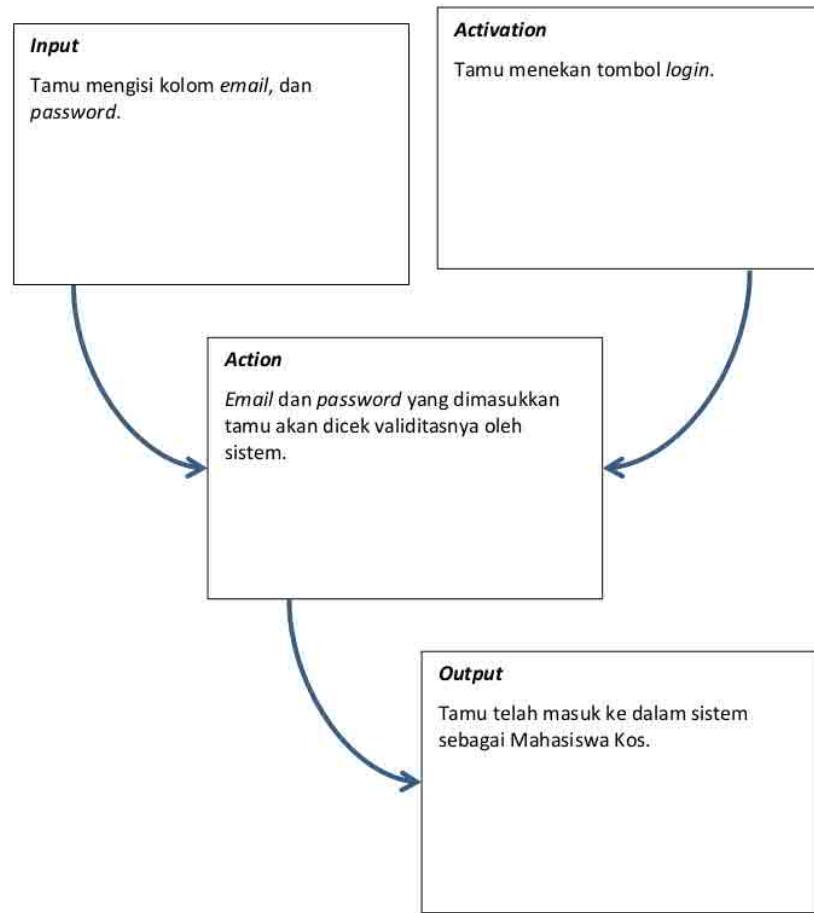
Fitur adalah sebuah cara yang memungkinkan pengguna mengakses dan menggunakan fungsionalitas produk, sehingga daftar fitur mendefinisikan fungsionalitas sistem secara keseluruhan. Berikut merupakan fitur-fitur yang mengacu pada *user story*.



Gambar 4.14 Fitur *register* pada *prototype* iterasi kedua

Pada gambar 4.14 terdapat fitur *register* yang mengacu pada *user story*.

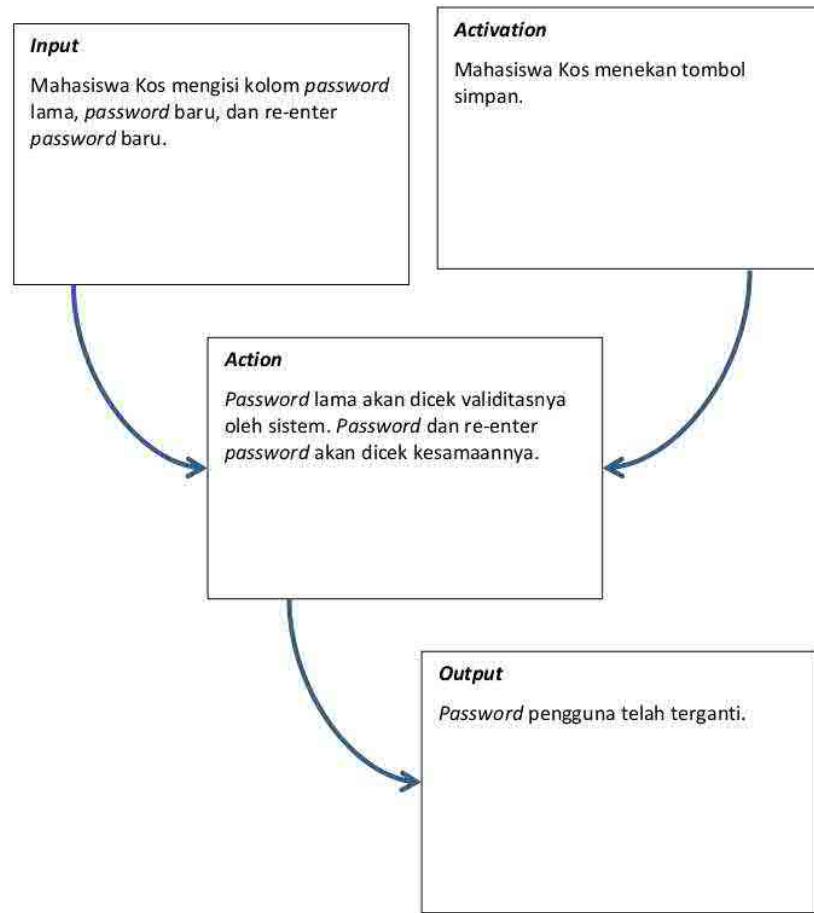
Fitur *login*



Gambar 4.15 Fitur *login* pada *prototype* iterasi kedua

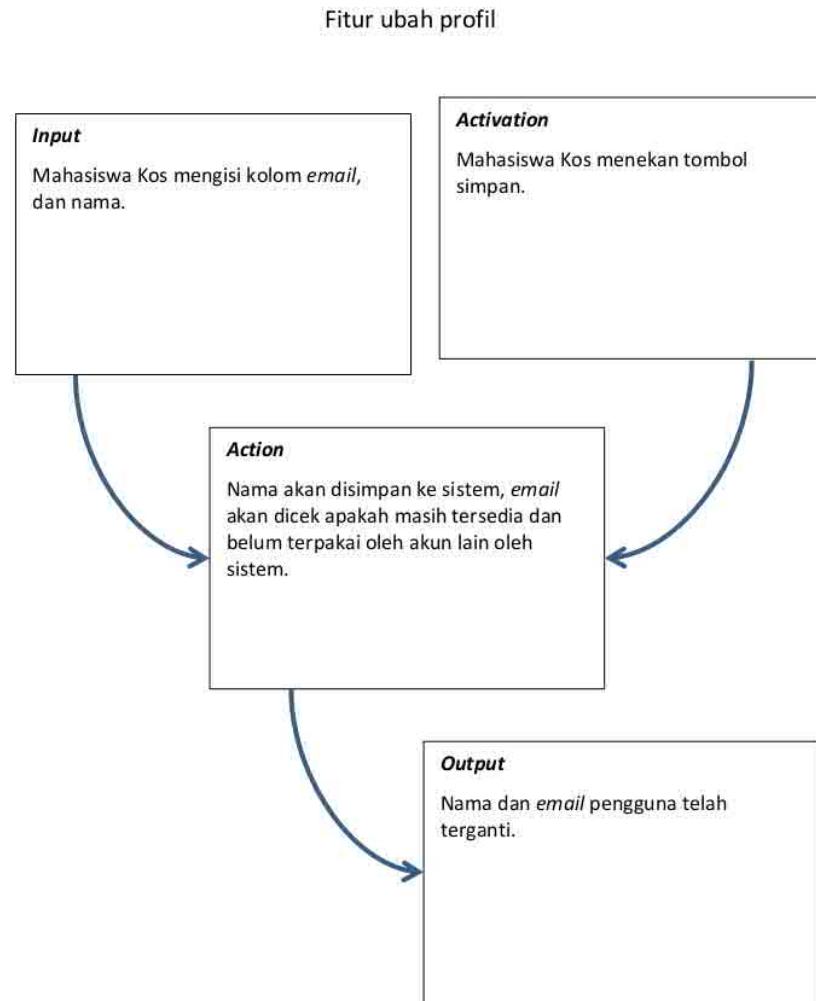
Pada gambar 4.15 terdapat fitur *login* yang mengacu pada *user story*.

Fitur ganti *password*



Gambar 4.16 Fitur ganti *password* pada *prototype* iterasi kedua

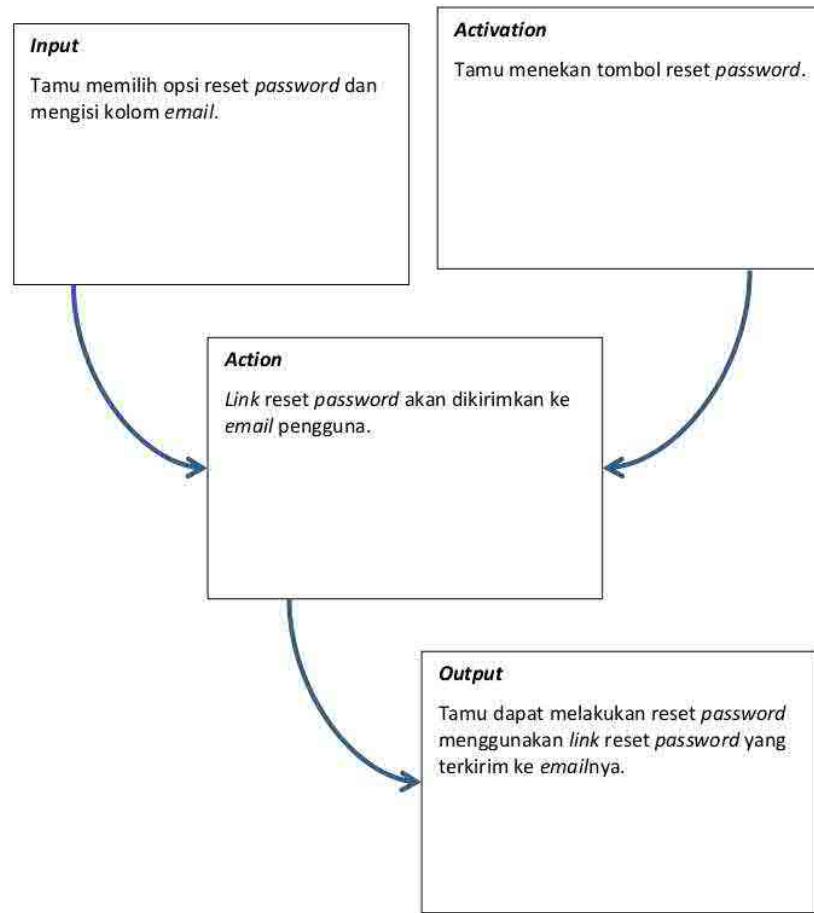
Pada gambar 4.16 terdapat fitur ganti *password* yang mengacu pada *user story*.



Gambar 4.17 Fitur ubah profil pada *prototype* iterasi kedua

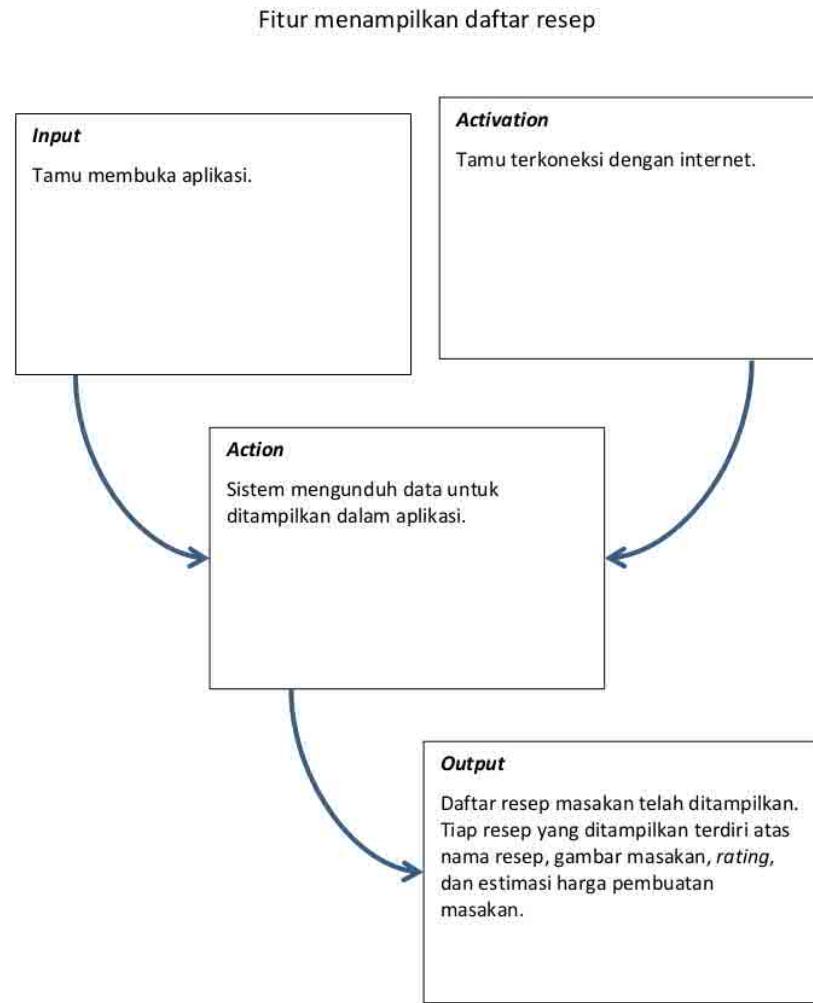
Pada gambar 4.17 terdapat fitur ubah profil yang mengacu pada *user story*.

Fitur reset *password*



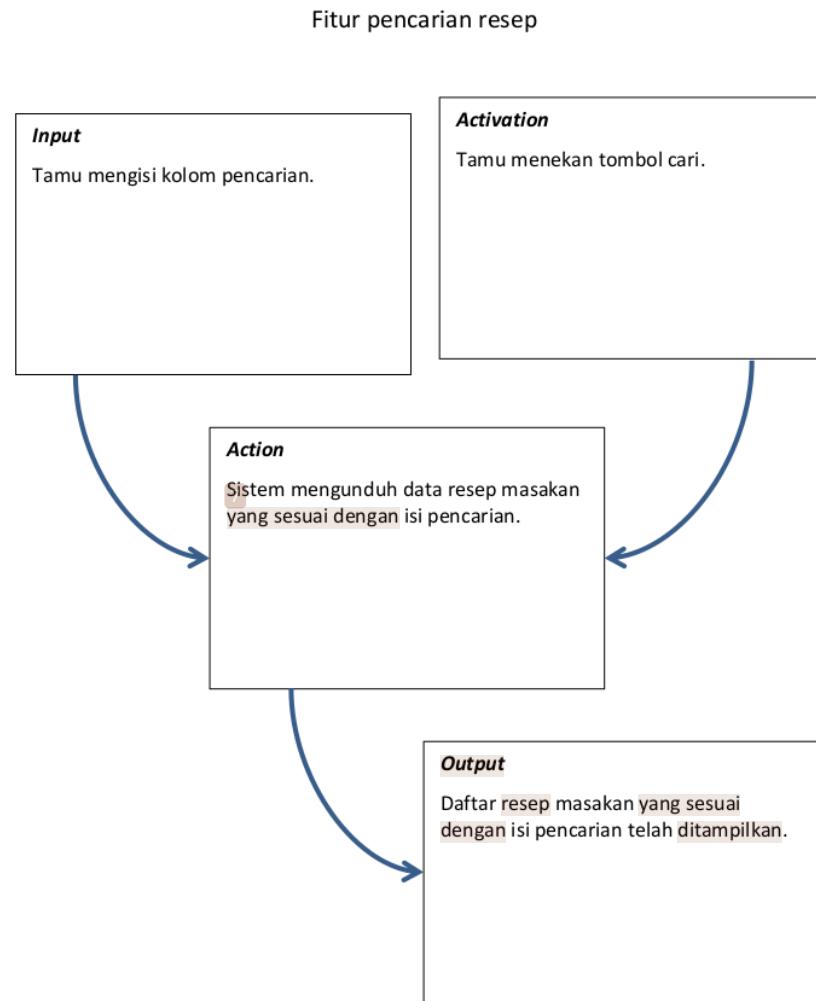
Gambar 4.18 Fitur reset *password* pada *prototype* iterasi kedua

Pada gambar 4.18 terdapat fitur reset *password* yang mengacu pada *user story*.



Gambar 4.19 Fitur menampilkan daftar resep pada *prototype* iterasi kedua

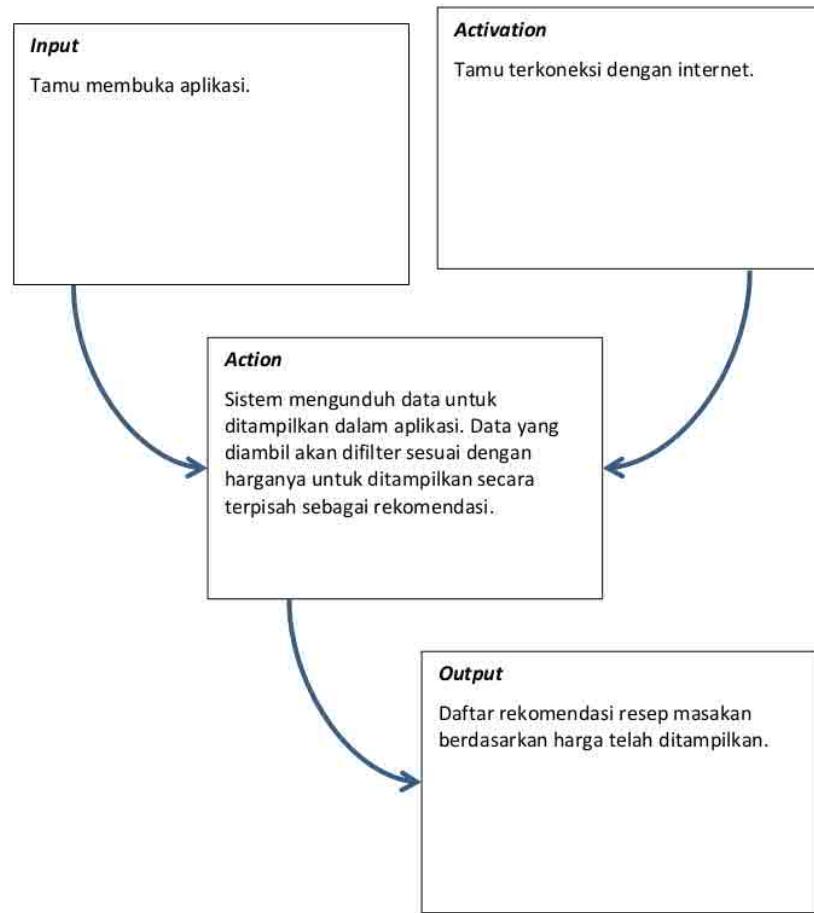
Pada gambar 4.19 terdapat fitur daftar resep yang mengacu pada *user story*.



Gambar 4.20 Fitur pencarian resep pada *prototype* iterasi kedua

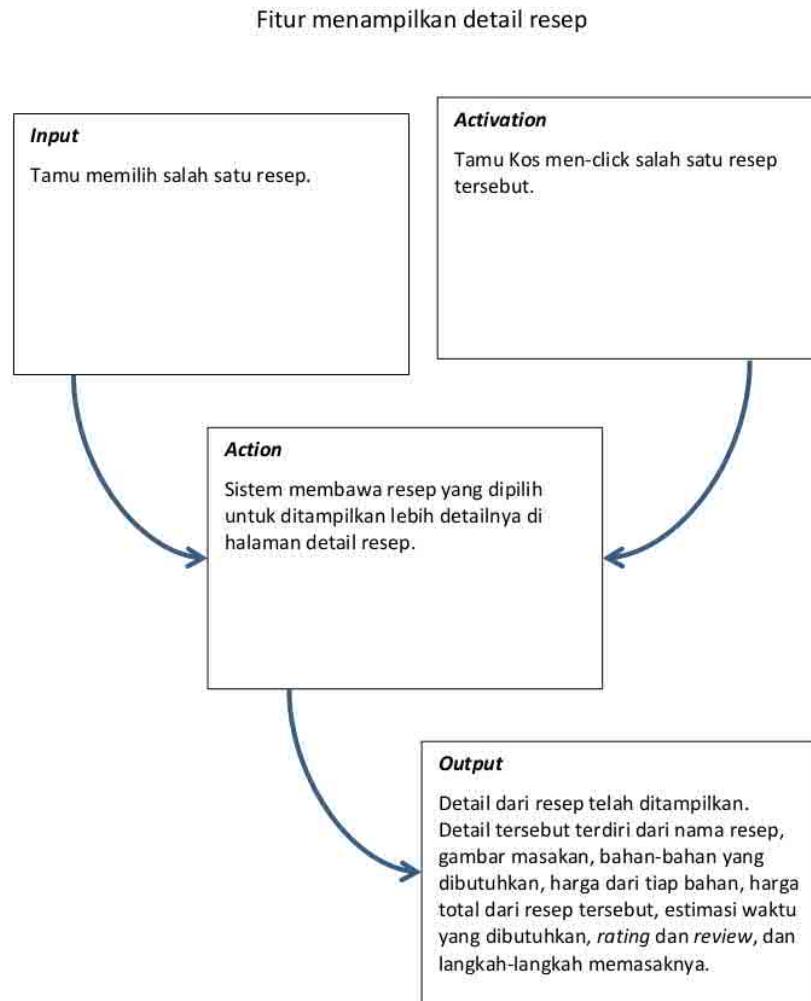
Pada gambar 4.20 terdapat fitur pencarian resep yang mengacu pada *user story*.

Fitur menampilkan rekomendasi resep masakan berdasarkan harga



Gambar 4.21 Fitur menampilkan rekomendasi resep masakan berdasarkan harga pada prototype iterasi kedua

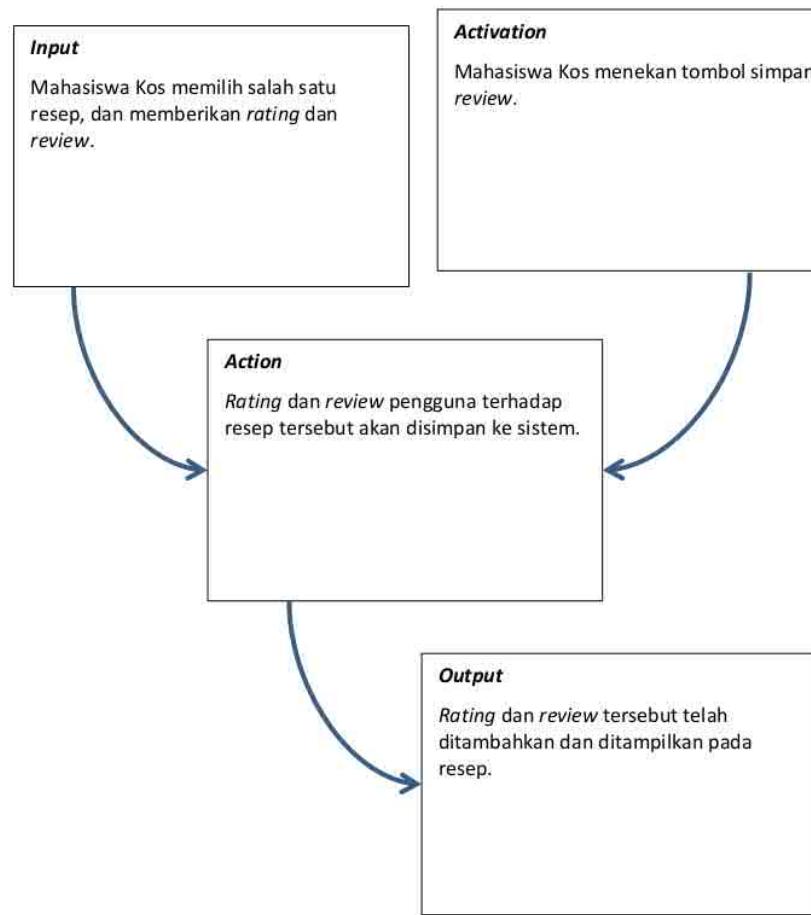
Pada gambar 4.21 terdapat fitur menampilkan rekomendasi resep masakan berdasarkan harga yang mengacu pada *user story*.



Gambar 4.22 Fitur menampilkan detail resep pada prototype iterasi kedua

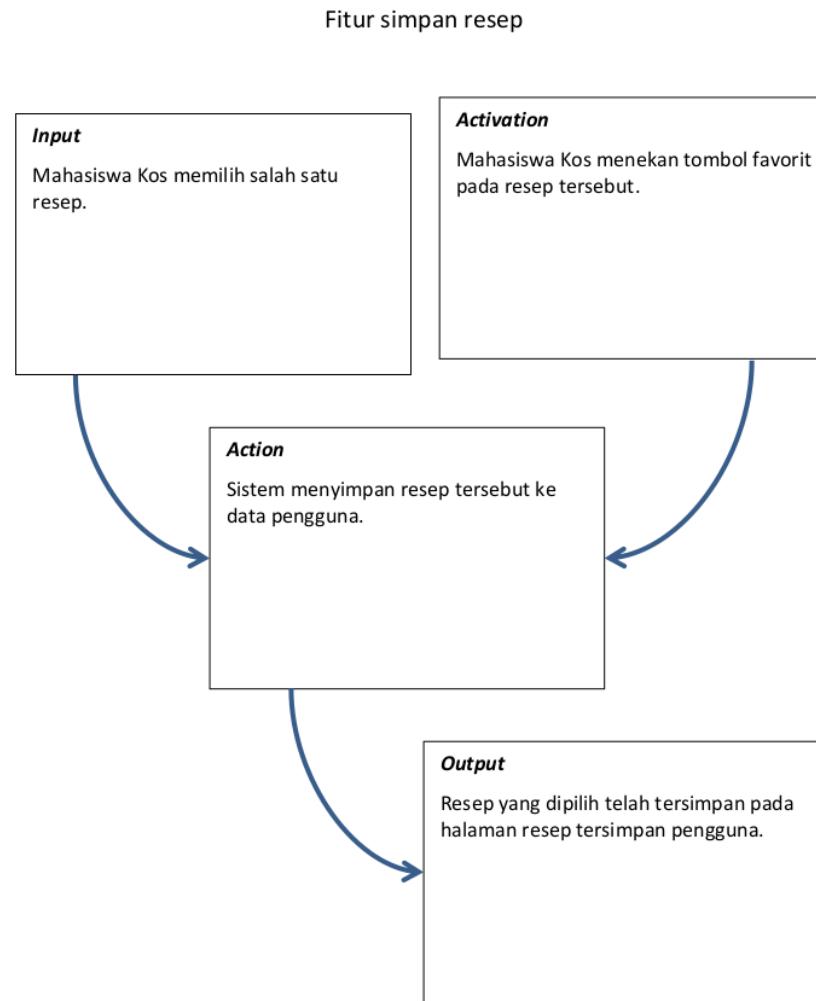
Pada gambar 4.22 terdapat fitur menampilkan detail resep yang mengacu pada *user story*.

Fitur menambahkan *rating* dan *review*



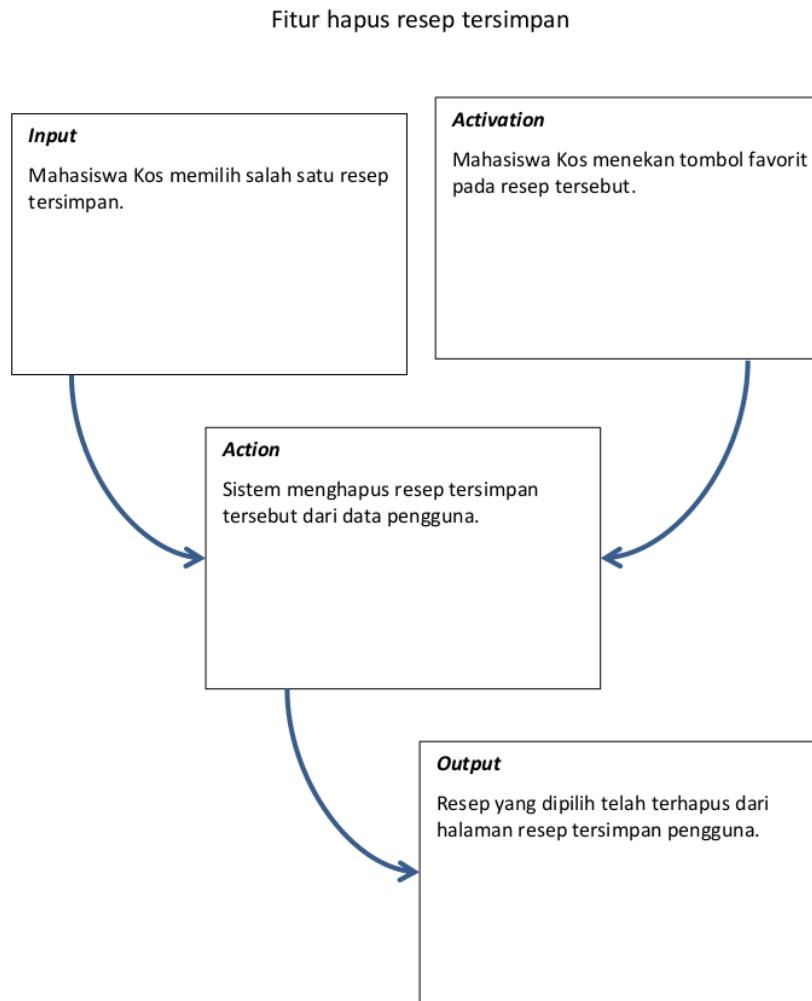
Gambar 4.23 Fitur menambahkan *rating* dan *review* pada prototype iterasi kedua

Pada gambar 4.23 terdapat fitur menambahkan *rating* dan *review* yang mengacu pada *user story*.



Gambar 4.24 Fitur simpan resep pada *prototype* iterasi kedua

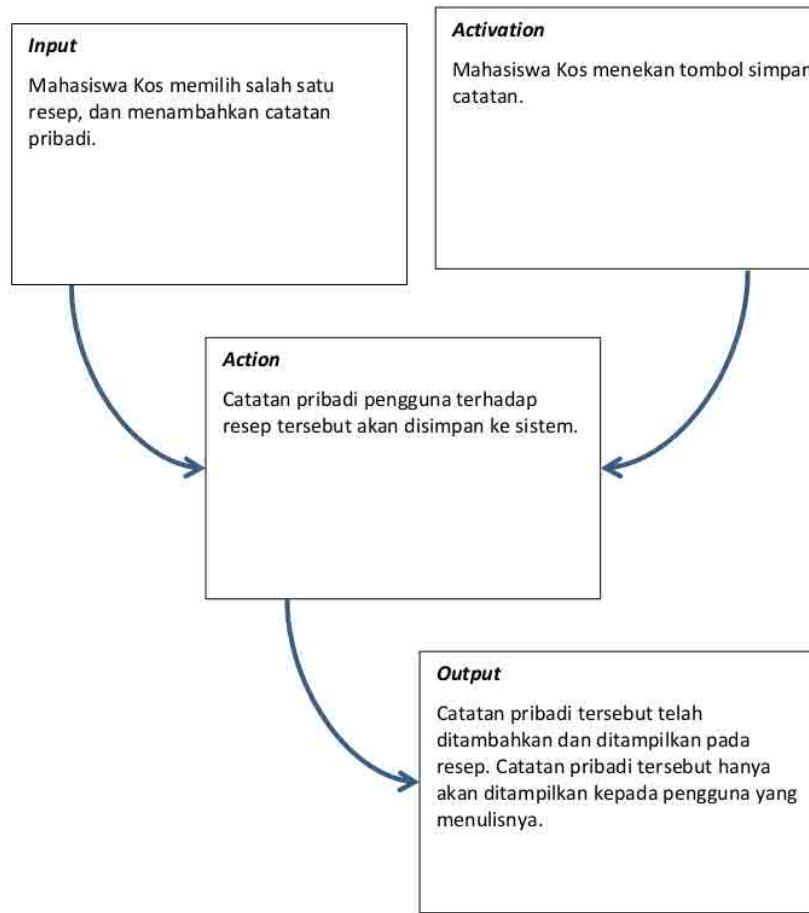
Pada gambar 4.24 terdapat fitur simpan resep yang mengacu pada *user story*.



Gambar 4.25 Fitur hapus resep tersimpan pada *prototype* iterasi kedua

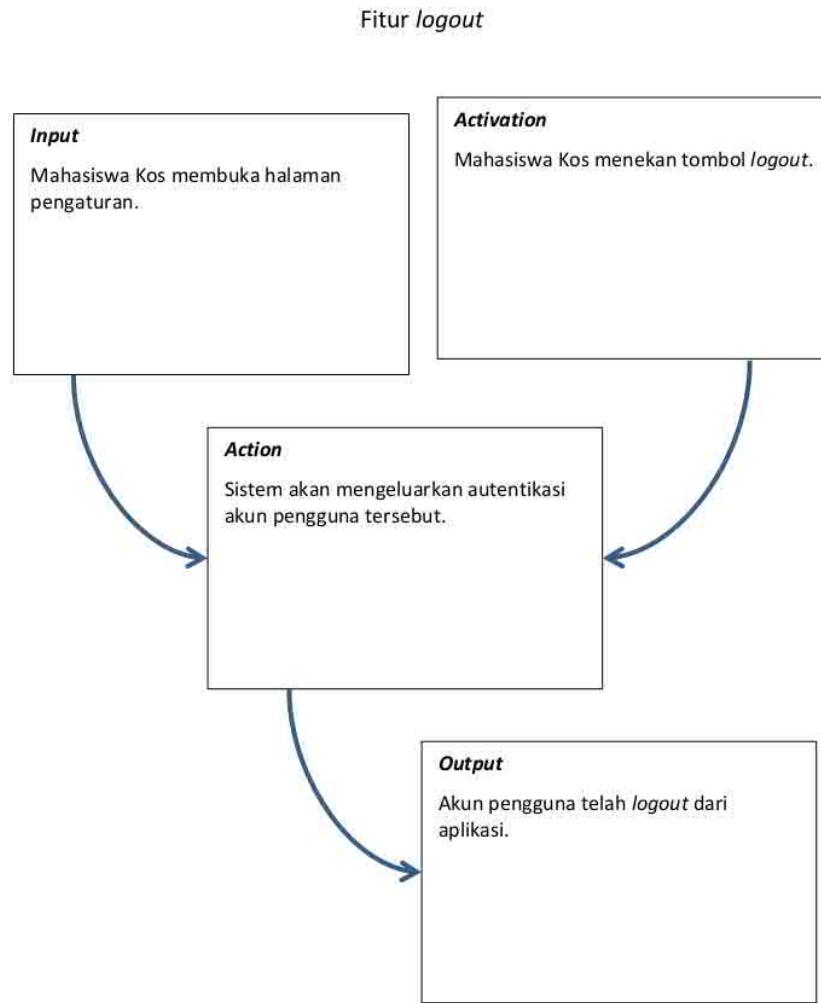
Pada gambar 4.25 terdapat fitur hapus resep tersimpan yang mengacu pada *user story*.

Fitur tambah catatan pribadi pada resep



Gambar 4.26 Fitur tambah catatan pribadi pada resep pada *prototype* iterasi kedua

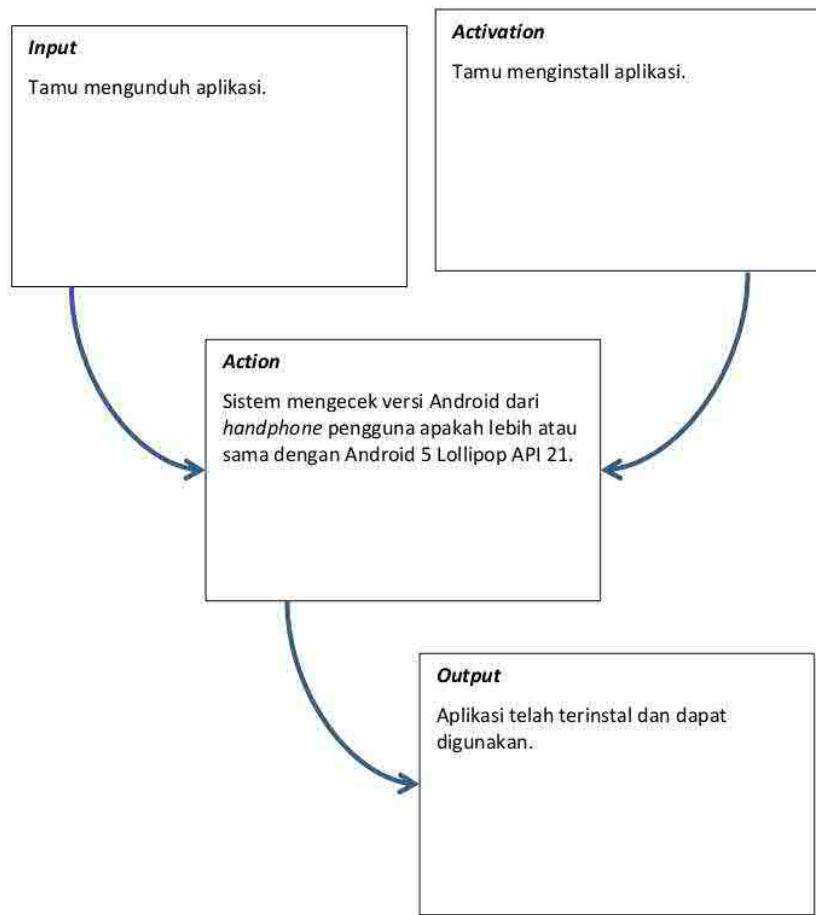
Pada gambar 4.26 terdapat fitur tambah catatan pribadi pada resep yang mengacu pada *user story*.



Gambar 4.27 Fitur *logout* pada prototype iterasi kedua

Pada gambar 4.27 terdapat fitur *logout* yang mengacu pada *user story*.

Fitur minimum Android 5 Lollipop API 21



Gambar 4.28 Fitur minimum Android 5 Lollipop API 21 pada *prototype* iterasi kedua

Pada gambar 4.28 terdapat fitur minimum Android 5 Lollipop API 21 yang mengacu pada *user story*.

4.3.4 Prototype

Pada subbab ini akan dijabarkan sebuah desain *prototype* dari aplikasi yang mengacu kepada *persona*, *scenario*, *user story*, dan fitur. *Prototype* tersebut sudah dapat diklik dan digunakan oleh pengguna, sehingga nantinya dapat dievaluasi kepada pengguna pada tahap evaluasi *prototype* iterasi pertama.

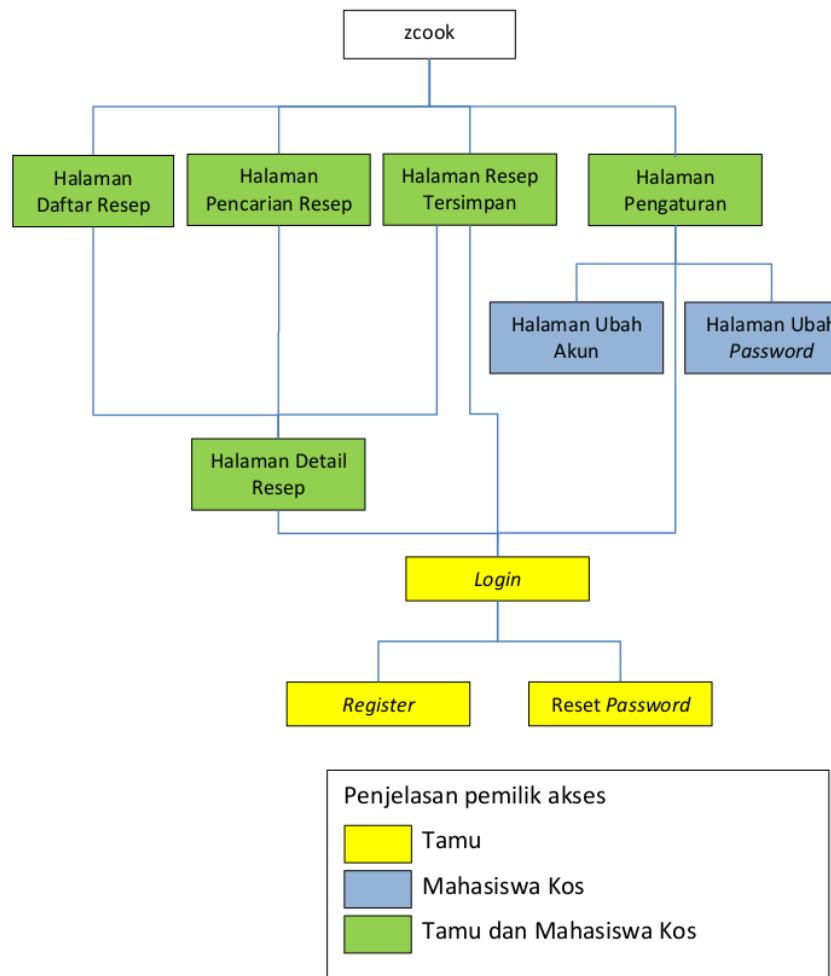
Pada aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa ini terdapat pengguna yang akan menjalankan sistem. Pengguna yang akan menggunakan sistem ini diuraikan pada tabel 4.8.

Tabel 4.8 Identifikasi pengguna pada *prototype* iterasi kedua

Jenis Pengguna	Deskripsi
Tamu	Tamu merupakan pengguna yang belum masuk ke dalam sistem. Tamu dapat mencari resep, melihat resep dari daftar resep yang tersedia, dan melakukan registrasi dan <i>login</i> .
Mahasiswa Kos	Mahasiswa Kos merupakan Tamu yang telah terdaftar dalam sistem. Mahasiswa Kos dapat mencari resep, melihat resep dari daftar resep yang tersedia, menyimpan resep, memberikan <i>review</i> dan <i>rating</i> , dan menambahkan catatan pribadi di tiap resep.

Pada tabel 4.8 terdapat beberapa jenis pengguna yang akan menggunakan sistem, yang mana terdiri dari Tamu dan Mahasiswa Kos.

Berikut merupakan *information architecture* sistem.

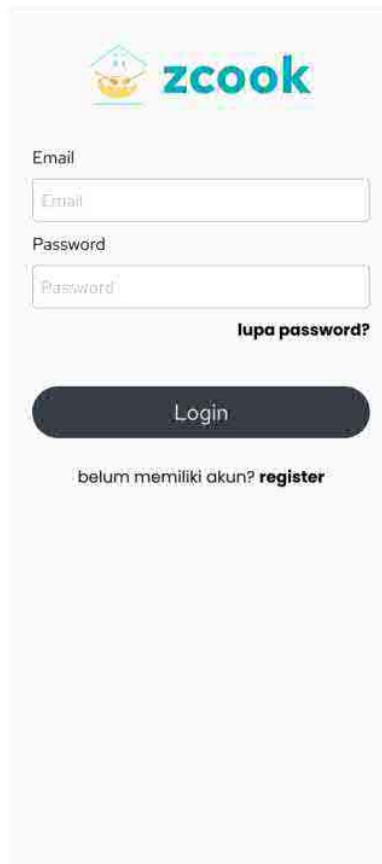


Gambar 4.29 Information architecture sistem pada prototype iterasi kedua

Pada gambar 4.29 terdapat information architecture sistem yang terdiri dari beberapa halaman, yang dikategorikan sesuai pemilik aksesnya.

4.3.4.1 Desain Halaman *Login*

Berikut merupakan desain halaman *login*.

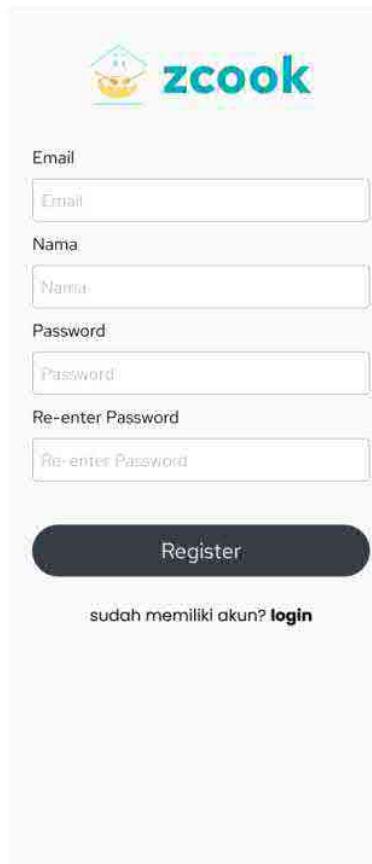


Gambar 4.30 Desain halaman *login* pada *prototype* iterasi kedua

Pada halaman *login*, terdapat kolom untuk mengisi *email* dan *password* yang nantinya akan diisi oleh pengguna, dan divalidasi oleh sistem saat tombol *login* ditekan. Terdapat pula teks *register* yang menavigasikan pengguna ke halaman *register*, dan teks *lupa password* yang menavigasikan pengguna ke halaman *reset password*.

4.3.4.2 Desain Halaman *Register*

Berikut merupakan desain halaman *register*.

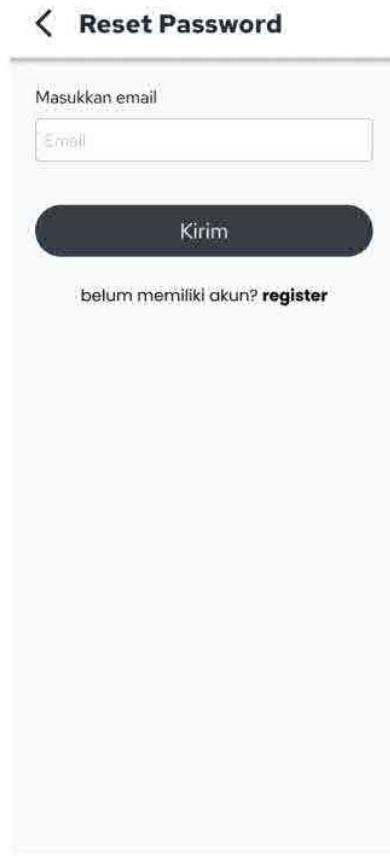


Gambar 4.31 Desain halaman *register* pada *prototype* iterasi kedua

Pada halaman *register*, terdapat kolom untuk mengisi *email*, *nama*, *password*, dan *re-enter password* yang nantinya akan diisi oleh pengguna, dan divalidasi oleh sistem saat tombol *register* ditekan. *Password* dan *re-enter password* akan dicek kesamaannya, dan *email* akan dicek apakah masih tersedia dan belum terpakai oleh akun lain oleh sistem. Terdapat pula teks *login* yang menavigasikan pengguna ke halaman *login*.

4.3.4.3 Desain Halaman Reset Password

Berikut merupakan desain halaman *register*.

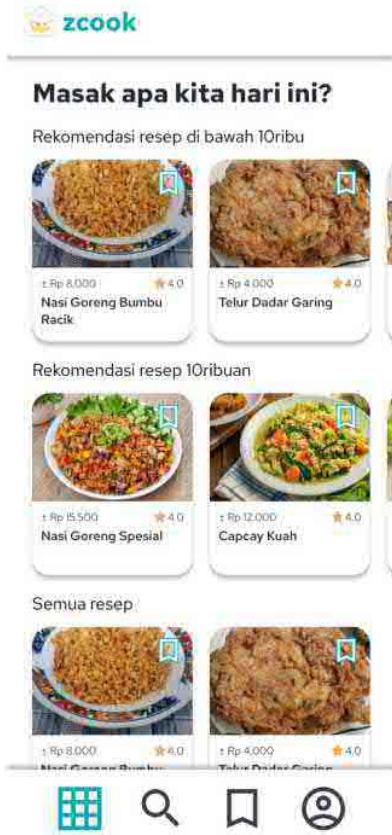


Gambar 4.32 Desain halaman reset *password* pada *prototype* iterasi kedua

Pada halaman reset *password*, terdapat kolom untuk mengisi *email* yang nantinya akan diisi oleh pengguna, dan menjadi tujuan pengiriman *link* reset *password* saat tombol *kirim* ditekan. Terdapat pula teks *register* yang menavigasikan pengguna ke halaman *register*, dan tombol *back* untuk kembali ke halaman sebelumnya.

4.3.4.4 Desain Halaman Daftar Resep

Berikut merupakan desain halaman daftar resep.

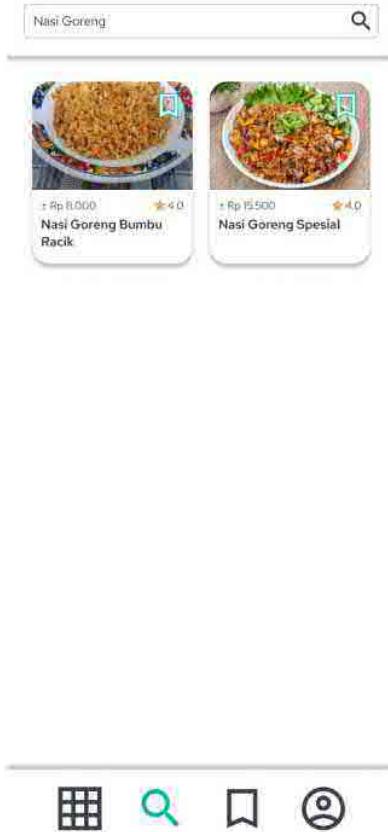


Gambar 4.33 Desain halaman daftar resep pada *prototype* iterasi kedua

Pada halaman daftar resep, terdapat rekomendasi resep sesuai harganya, dalam hal ini resep 10ribuan dan resep di bawah 10ribu, dan juga terdapat daftar semua resep yang tersedia. Tiap resep yang ditampilkan terdiri atas nama resep, gambar masakan, icon simpan resep, *rating*, dan estimasi harga pembuatan masakan. Jika pengguna belum masuk, maka icon simpan resep saat diklik akan navigasikan pengguna ke halaman *login*, dan jika sudah masuk maka icon simpan resep saat diklik akan menyimpan resep tersebut ke daftar resep. Resep yang ditekan akan navigasikan pengguna ke halaman detail dari resep tersebut. Terdapat juga tombol pencarian yang akan navigasikan pengguna ke halaman pencarian, dan *bottom navigation bar* atau bilah navigasi bawah yang terdiri dari menu daftar resep, resep tersimpan, dan pengaturan yang akan navigasikan masing-masing ke halaman daftar resep, halaman resep tersimpan, dan halaman pengaturan.

4.3.4.5 Desain Halaman Pencarian Resep

Berikut merupakan desain halaman pencarian resep.



Gambar 4.34 Desain halaman pencarian resep pada prototype iterasi kedua

Pada halaman pencarian resep, terdapat kolom untuk mengisi kata kunci pencarian, tombol cari untuk melakukan pencarian, dan tombol back untuk kembali ke halaman sebelumnya. Terdapat pula daftar hasil pencarian resep. Tiap resep yang ditampilkan terdiri atas nama resep, gambar masakan, icon simpan resep, rating, dan estimasi harga pembuatan masakan. Jika pengguna belum masuk, maka icon simpan resep saat diklik akan navigasikan pengguna ke halaman login, dan jika sudah masuk maka icon simpan resep saat diklik akan menyimpan resep tersebut ke daftar resep. Resep yang ditekan akan navigasikan pengguna ke halaman detail dari resep tersebut.

4.3.4.6 Desain Halaman Detail Resep Tamu

Berikut merupakan desain halaman login.



Nasi Goreng Spesial

★ 4.0

Total harga : ± Rp 15.500

Estimasi waktu masak : ± 20 menit

Bahan :

-Nasi putih 1 piring	± Rp 3.000
-Bawang putih 2 siung	± Rp 500
-Kecap manis 1 sdm	± Rp 1.000 (20ml)
-Saus sambal 1 sdm	± Rp 500 (10gr)
-Saus tiram 1 sdm	± Rp 3.000 (23ml)
-Garam 1 sdt	± Rp 2.000 (200gr)
-Penyedap rasa 1 sdt	± Rp 500 (8gr)
-Daun bawang 1 batang	± Rp 1.000 (80gr)
-Telur ayam 1 butir	± Rp 2.000
-Sosis ayam 1 buah, iris	± Rp 1.000
-Minyak goreng 3 sdm	± Rp 1000 (3 sdm)

Cara membuat :

1. Siapkan penggorengan dengan api sedang, tuang margarin atau minyak goreng.
2. Masukkan bawang putih dan daun bawang yang sudah dicincang halus. Tumis hingga berbau harum atau hingga warnanya keemasan.
3. Masukkan sosis dan 1 butir telur ayam. Tumis sebentar.
4. Masukkan bumbu halus dan nasi. Aduk hingga tercampur rata.
5. Tuang kecap manis, saus sambal, saus tiram, garam, dan kaldu bubuk. Aduk hingga warna nasi berubah secara merata.
6. Nasi goreng biasa yang sederhana, dan enak siap disajikan.



Masuk ke akunmu

Masuk agar dapat simpan resep dan review.

Kata orang-orang tentang resep ini

Aldi



lumayan

Gambar 4.35 Desain halaman detail resep Tamu pada prototype iterasi kedua

Pada halaman detail resep, terdapat nama resep, gambar masakan, bahan-bahan yang dibutuhkan, harga dari tiap bahan, harga total dari resep tersebut, estimasi waktu yang dibutuhkan, rating dan review, dan langkah-langkah memasaknya. Terdapat juga tombol kembali untuk kembali ke halaman sebelumnya, icon simpan resep, dan tombol masuk ke akun yang akan menavigasikan pengguna ke halaman login. Jika diklik, icon simpan resep akan menavigasikan pengguna ke halaman login.

4.3.4.7 Desain Halaman Detail Resep Mahasiswa Kos

Berikut merupakan desain halaman login.



Nasi Goreng Spesial

★ 4.0

Total harga : ± Rp 15.500

Estimasi waktu masak : ± 20 menit

Bahan :

-Nasi putih 1 piring	± Rp 3.000
-Bawang putih 2 siung	± Rp 500
-Kecap manis 1 sdm	± Rp 1.000 (20ml)
-Saus sambal 1 sdm	± Rp 500 (10gr)
-Saus tiram 1 sdm	± Rp 3.000 (23ml)
-Garam 1 sdt	± Rp 2.000 (200gr)
-Penyedap rasa 1 sdt	± Rp 500 (8gr)
-Daun bawang 1 batang	± Rp 1.000 (80gr)
-Telur ayam 1 butir	± Rp 2.000
-Sosis ayam 1 buah, iris	± Rp 1.000
-Minyak goreng 3 sdm	± Rp 1000 (3 sdm)

Cara membuat :

- Siapkan penggorengan dengan api sedang, tuang margarin atau minyak goreng.
- Masukkan bawang putih dan daun bawang yang sudah dicincang halus. Tumis hingga berbau harum atau hingganya keemasan.

- Masukkan sosis dan 1 butir telur ayam. Tumis sebentar.
- Masukkan bumbu halus dan nasi. Aduk hingga tercampur rata.
- Tuang kecap manis, saus sambal, saus tiram, garam, dan kaldu bubuk. Aduk hingga warna nasi berubah secara merata.
- Nasi goreng biasa yang sederhana, dan enak siap disajikan.

Catatan pribadi

nota yang mau dicatat di resep ini

Simpan catatan

Review resep ini!



ginaiini iku resipi ni?

Simpan review

Kata orang-orang tentang resep ini

Aldi
★★★★★

lumayan

Gambar 4.36 Desain halaman detail resep Mahasiswa Kos pada prototype iterasi kedua

Pada halaman detail resep, terdapat nama resep, gambar masakan, bahan-bahan yang dibutuhkan, harga dari tiap bahan, harga total dari resep tersebut, estimasi waktu yang dibutuhkan, rating dan review, dan langkah-langkah memasaknya. Terdapat kolom untuk mengisi catatan pribadi dan tombol untuk menyimpan catatan, bintang untuk memberikan rating beserta kolom untuk mengisi review dan tombol untuk menyimpan review, icon simpan resep, dan tombol kembali untuk kembali ke halaman sebelumnya. Jika diklik, icon simpan resep akan menyimpan resep tersebut ke daftar resep.

4.3.4.8 Desain Halaman Resep Tersimpan Tamu

Berikut merupakan desain halaman login.

Resep Tersimpan



Gambar 4.37 Desain halaman resep tersimpan Tamu pada prototype iterasi kedua

Pada halaman resep tersimpan, terdapat tombol masuk ke akun yang akan menavigasikan pengguna ke halaman *login*. Terdapat juga *bottom navigation bar* atau bilah navigasi bawah yang terdiri dari menu daftar resep, resep tersimpan, dan pengaturan yang akan menavigasikan masing-masing ke halaman daftar resep, halaman resep tersimpan, dan halaman pengaturan.

4.3.4.9 Desain Halaman Resep Tersimpan Mahasiswa Kos

Berikut merupakan desain halaman *login*.

Resep Tersimpan



Gambar 4.38 Desain halaman resep tersimpan Mahasiswa Kos pada prototype iterasi kedua

Pada halaman resep tersimpan, terdapat daftar resep yang telah kita simpan. Tiap resep yang ditampilkan terdiri atas nama resep, gambar masakan, icon simpan resep, rating, dan estimasi harga pembuatan masakan. Jika icon simpan resep diklik, maka resep tersebut akan dihapus dari daftar resep tersimpan. Resep yang ditekan akan menavigasikan pengguna ke halaman detail dari resep tersebut. Terdapat juga *bottom navigation bar* atau bilah navigasi bawah yang terdiri dari menu daftar resep, resep tersimpan, dan pengaturan yang akan menavigasikan masing-masing ke halaman daftar resep, halaman resep tersimpan, dan halaman pengaturan.

4.3.4.10 Desain Halaman Pengaturan Tamu

Berikut merupakan desain halaman pengaturan tamu.

Pengaturan



Gambar 4.39 Desain halaman pengaturan Tamu pada prototype iterasi kedua

Pada halaman pengaturan Tamu, terdapat tombol masuk ke akun yang akan menavigasikan pengguna ke halaman *login*. Terdapat juga *bottom navigation bar* atau bilah navigasi bawah yang terdiri dari menu daftar resep, resep tersimpan, dan pengaturan yang akan menavigasikan masing-masing ke halaman daftar resep, halaman resep tersimpan, dan halaman pengaturan.

4.3.4.11 Desain Halaman Pengaturan Mahasiswa Kos

Berikut merupakan desain halaman pengaturan Mahasiswa Kos.

Pengaturan



Gambar 4.40 Desain halaman pengaturan Mahasiswa Kos pada *prototype* iterasi kedua

Pada halaman pengaturan, terdapat nama pengguna, *email* pengguna, tombol ubah akun, tombol ubah *password*, dan tombol keluar akun. Tombol ubah akun akan menavigasikan pengguna ke halaman ubah akun, tombol ubah *password* akan menavigasikan pengguna ke halaman ubah *password*, dan tombol keluar akun akan mengeluarkan pengguna dari hak akses pengguna Mahasiswa Kos dan menavigasikan pengguna ke halaman *login*. Terdapat juga *bottom navigation bar* atau bilah navigasi bawah yang terdiri dari menu daftar resep, resep tersimpan, dan pengaturan yang akan menavigasikan masing-masing ke halaman daftar resep, halaman resep tersimpan, dan halaman pengaturan.

4.3.4.12 Desain Halaman Ubah Akun

Berikut merupakan desain halaman ubah akun.

Ubah Akun

Nama
Muhammad Ziyad Al Maududi

Email
mziyad195@gmail.com

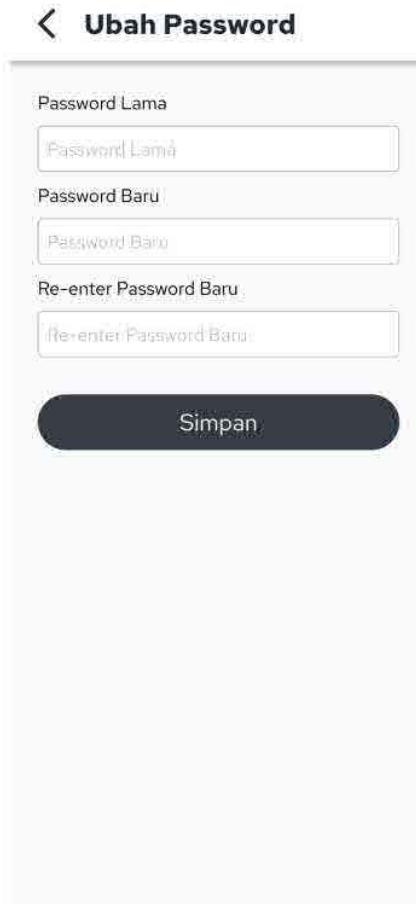
Simpan

Gambar 4.41 Desain halaman ubah akun Mahasiswa Kos pada *prototype* iterasi kedua

Pada halaman ubah akun, terdapat kolom untuk mengisi nama, dan email, yang nantinya akan diisi oleh pengguna, dan diproses untuk disimpan oleh sistem saat tombol simpan ditekan. Terdapat juga tombol kembali, untuk kembali ke halaman sebelumnya.

4.3.4.13 Desain Halaman Ubah Password

Berikut merupakan desain halaman ubah password.



Gambar 4.42 Desain halaman ubah password Mahasiswa Kos pada prototype iterasi kedua

Pada halaman ubah password, terdapat kolom untuk mengisi password lama, password baru, dan re-enter password baru, yang nantinya akan diisi oleh pengguna, dan diproses untuk disimpan oleh sistem saat tombol simpan ditekan. Password lama akan dicek oleh sistem validitasnya, dan password dan re-enter password akan dicek kesamaannya. Terdapat juga tombol kembali, untuk kembali ke halaman sebelumnya.

4.3.5 Evaluasi Prototype

Setelah prototype yang telah dibuat diuji kepada pengguna dan dilakukan evaluasi, tidak terdapat perbaikan atau penambahan fitur dan fungsionalitas perangkat lunak. Hasil dari evaluasi prototype iterasi kedua dapat dilihat pada Lampiran C.2.

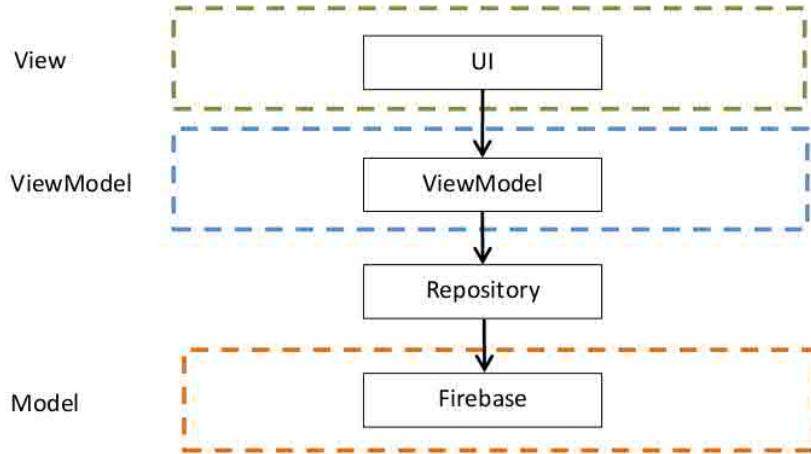
BAB 5 PERANCANGAN

5.1 Rancangan Arsitektur Sistem

21

Pengembangan perangkat lunak pada penelitian ini menggunakan arsitektur MVVM atau Model-View-ViewModel, dan menggunakan Firebase sebagai *backend* atau tempat penyimpanan data.

Berikut merupakan alur pengelolaan data pada perangkat lunak.



Gambar 5.1 Alur pengelolaan data MVVM Android pada aplikasi

Sumber: Adaptasi Arif, Musthafa, & Muriyatmoko (2019)

Pada gambar 5.1 terdapat penjabaran lebih detail terkait arsitektur MVVM Android yang diadaptasi dari gambar 2.37. Model mewakili proses logika bisnis dan data yang menentukan bagaimana data dimanipulasi. Untuk Model sendiri diwakili oleh Firebase yang akan menjadi sumber data yang akan dikelola oleh Repository. Repository akan mendistribusikan data ke ViewModel. ViewModel merupakan pengontrol yang mendukung koordinasi dan interaksi antara View dan Model. View merupakan komponen aktif yang berisi perilaku, peristiwa, dan pengikatan data, yang mana membutuhkan informasi mengenai komponen ViewModel, dan Model yang mendasarinya.

Penjelasan lebih lanjut terkait komponen-komponen yang terdapat pada gambar 5.1 yaitu sebagai berikut:

A. UI merepresentasikan tampilan antarmuka yang berhubungan langsung dengan pengguna dan dapat berisi Activity dan Fragment.

B. ViewModel merupakan sebuah kelas yang membantu mempertahankan state tampilan, memanipulasi Model yang dilakukan pada View, dan memicu peristiwa dalam View itu sendiri.

C. Repository merupakan sebuah kelas yang berisi fungsi-fungsi yang mengatur distribusi dan pengelolaan data pada aplikasi, yang mana pada aplikasi ini berasal dari Firebase.

D. Firebase merupakan sebuah layanan *back-end* yang digunakan dalam penyimpanan dan pengelolaan data pada server di internet.

Berikut merupakan skema arsitektur Firebase.



Gambar 5.2 Skema arsitektur Firebase pada aplikasi

Sumber: Adaptasi GeeksForGeeks (2021)

Pada gambar 5.2 terdapat skema arsitektur Firebase yang terhubung dengan aplikasi ini, yang mana diadopsi dari gambar 2.36.

1

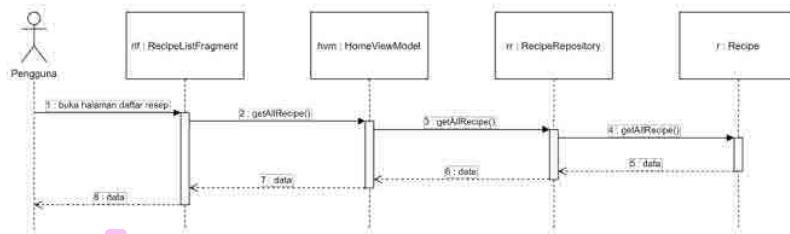
5.2 Rancangan *Sequence Diagram*

Sequence adalah sebuah diagram yang digunakan dalam penggambaran perilaku dari sebuah objek dengan memberikan peran dan pesan yang saling dikirim dan diterima antar objek, serta memodelkan urutan kejadian yang terjadi pada satu eksekusi sistem. Pada bagian ini akan dijabarkan beberapa sequence diagram yang menggambarkan fungsionalitas utama sistem yang didapat dari bab sebelumnya.

2

5.2.1 Sequence Diagram Menampilkan Daftar Resep

Sequence diagram menampilkan daftar resep menggambarkan perilaku, proses, dan interaksi antar objek dalam menampilkan daftar resep pada aplikasi.



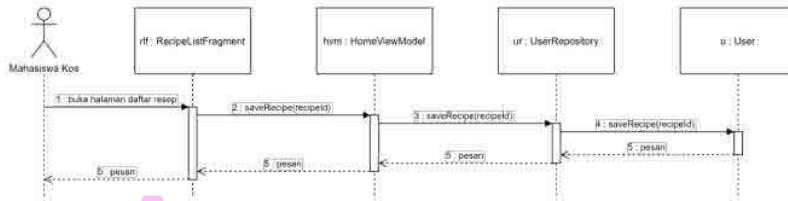
Gambar 5.3 *Sequence diagram* menampilkan daftar

Pada gambar 5.3 terdapat *sequence diagram* menampilkan daftar resep yang dimulai dari pengguna membuka halaman daftar resep, hingga data resep tersebut diterima oleh pengguna.

2

5.2.2 Sequence Diagram Simpan Resep

Sequence diagram simpan resep menggambarkan perilaku, proses, dan interaksi antar objek dalam melakukan simpan resep pada aplikasi.

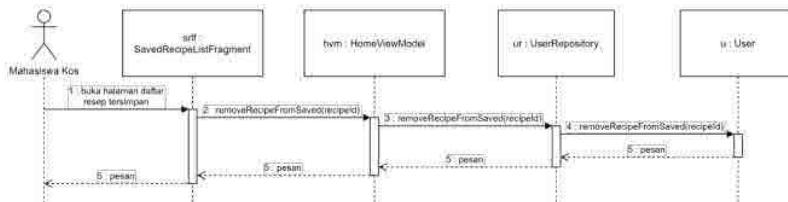


Gambar 5.4 *Sequence diagram* simpan resep

Pada gambar 5.4 terdapat *sequence diagram* simpan resep yang dimulai dari pengguna membuka halaman daftar resep, hingga resep tersebut tersimpan pada data pengguna, dalam hal ini User, dan kemudian pesan keberhasilan dari penyimpanan tersebut diterima oleh pengguna.

5.2.3 Sequence Diagram Hapus Resep Tersimpan

Sequence diagram hapus resep tersimpan menggambarkan perilaku, proses, dan interaksi antar objek dalam melakukan hapus resep tersimpan pada aplikasi.



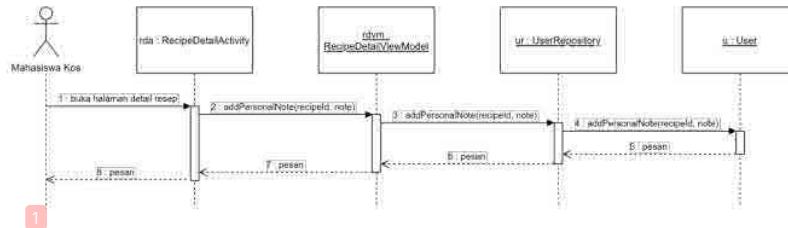
Gambar 5.5 *Sequence diagram* hapus resep tersimpan

Pada gambar 5.5 terdapat *sequence diagram* hapus resep tersimpan yang dimulai dari pengguna membuka halaman daftar resep tersimpan, hingga resep tersebut dihapus dari daftar resep tersimpan pada data pengguna, dalam hal ini User, dan kemudian pesan keberhasilan dari penyimpanan tersebut diterima oleh pengguna.

2

5.2.4 Sequence Diagram Tambah Catatan Pribadi Pada Resep

Sequence diagram tambah catatan pribadi pada resep menggambarkan perilaku, proses, dan interaksi antar objek dalam melakukan tambah catatan pribadi pada resep pada aplikasi.



Gambar 5.6 Sequence diagram tambah catatan pribadi pada resep

Pada gambar 5.6 terdapat *sequence diagram* tambah catatan pribadi pada resep yang dimulai dari pengguna membuka halaman detail resep, hingga pesan keberhasilan dari penambahan tersebut diterima oleh pengguna.

5.3 Rancangan Database

Entitas *database* yang terdapat pada pengembangan perangkat lunak ini yaitu MahasiswaKos, Recipe, PersonalNote, dan Review.

5.3.1 Rancangan Database Entitas MahasiswaKos

Entitas MahasiswaKos merupakan sebuah entitas yang digunakan untuk menyimpan informasi pengguna yang telah melakukan *login* terkait nama, resep yang disimpan, dan catatan pribadi dari tiap resep.

Tabel 5.1 Entitas MahasiswaKos

No	Nama Atribut	Tipe Data	Keterangan
1.	id	String	Identitas pembeda antar entitas MahasiswaKos
2.	name	String	Nama dari MahasiswaKos
3.	listSavedRecipId	List<String>	Daftar id dari resep yang disimpan oleh MahasiswaKos
4.	listPersonalNote	List<PersonalNote>	Daftar PersonalNote yang merupakan catatan pribadi akan suatu resep

Pada tabel 5.1 terdapat entitas MahasiswaKos beserta atribut yang terdapat di dalamnya.

5.3.2 Rancangan Database Entitas Recipe

Entitas Recipe merupakan sebuah entitas yang digunakan untuk menyimpan informasi resep, dan daftar *review* terhadap resep tersebut.

Tabel 5.2 Entitas Recipe

No	Nama Atribut	Tipe Data	Keterangan

1.	<code>id</code>	<code>String</code>	Identitas pembeda antar entitas Recipe
2.	<code>name</code>	<code>String</code>	Nama dari resep
3.	<code>imageUrl</code>	<code>String</code>	Url gambar dari resep
4.	<code>estimatedPrice</code>	<code>Int</code>	Estimasi biaya yang dibutuhkan dalam membuat sebuah resep
5.	<code>estimatedTime</code>	<code>String</code>	Estimasi waktu yang dibutuhkan dalam membuat sebuah resep
6.	<code>listIngredient</code>	<code>String</code>	Daftar bahan-bahan yang dibutuhkan dari sebuah resep
7.	<code>listIngredientPrice</code>	<code>String</code>	Daftar harga dari bahan-bahan yang dibutuhkan dari sebuah resep
8.	<code>steps</code>	<code>String</code>	Langkah-langkah pembuatan sebuah resep
9.	<code>listReview</code>	<code>List<Review></code>	Daftar <i>Review</i> atau ulasan dari sebuah resep

Pada tabel 5.2 terdapat entitas Recipe beserta atribut yang terdapat di dalamnya.

5.3.3 Rancangan Database Entitas PersonalNote

Entitas PersonalNote merupakan sebuah entitas yang digunakan untuk menyimpan informasi catatan pribadi MahasiswaKos akan sebuah Recipe.

Tabel 5.3 Entitas PersonalNote

No	Nama Atribut	Tipe Data	Keterangan
1.	<code>recipeId</code>	<code>String</code>	Id dari sebuah Recipe
2.	<code>note</code>	<code>String</code>	Catatan dari sebuah Recipe

Pada tabel 5.3 terdapat entitas PersonalNote beserta atribut yang terdapat di dalamnya.

5.3.4 Rancangan Database Entitas Review

Entitas Review merupakan sebuah entitas yang digunakan untuk menyimpan informasi *review* atau ulasan dari sebuah Recipe.

Tabel 5.4 Entitas Review

No	Nama Atribut	Tipe Data	Keterangan

1.	userId	String	Id dari MahasiswaKos yang membuat <i>Review</i>
2.	userName	String	Nama dari MahasiswaKos yang membuat <i>Review</i>
3.	<i>rating</i>	Double	<i>Rating</i> atau penilaian dengan skala angka
4.	<i>review</i>	String	Isi dari <i>review</i> atau ulasan

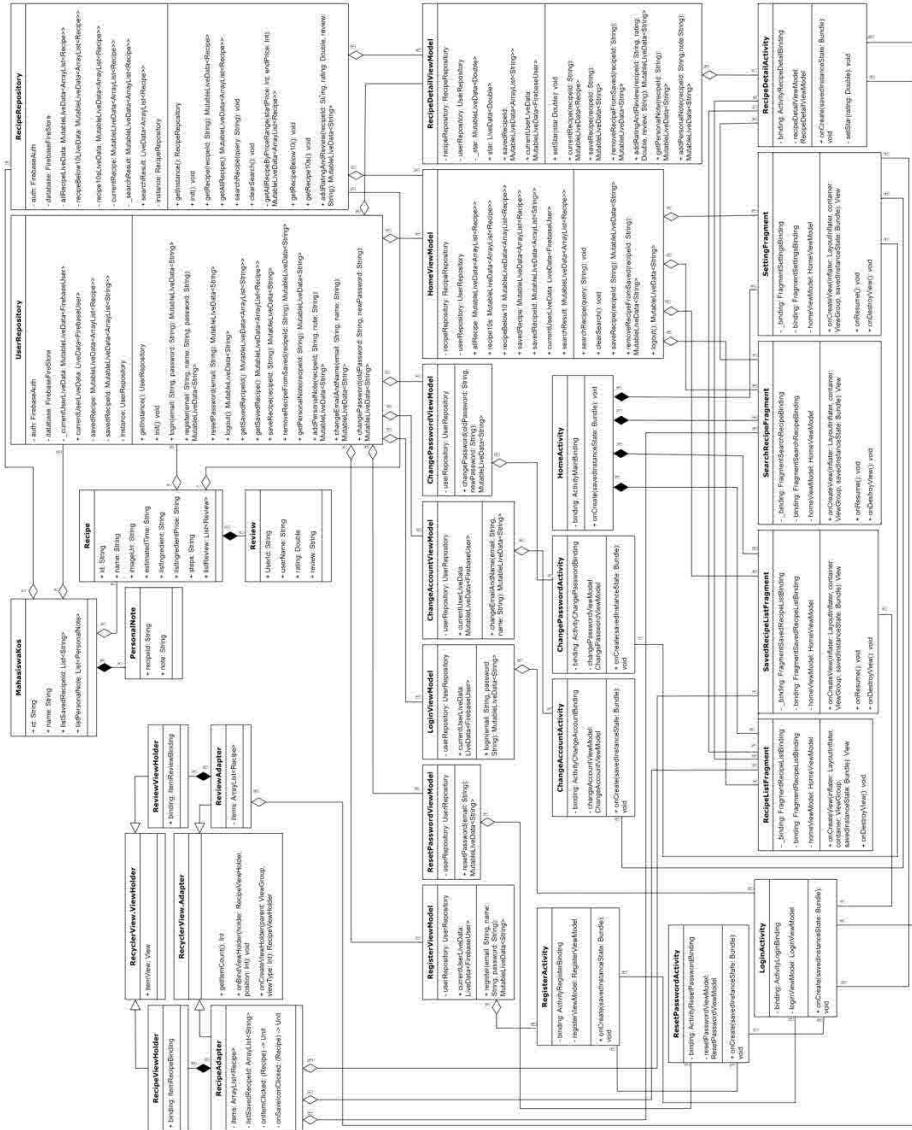
Pada tabel 5.4 terdapat entitas *Review* beserta atribut yang terdapat di dalamnya.

1

5.4 Rancangan *Class Diagram*

Class Diagram adalah sebuah diagram yang digunakan dalam menggambarkan struktur sebuah sistem dari sisi pendefinisian kelas-kelas yang membangun sistem beserta relasi di dalamnya, dimana setiap kelas terdiri atas atribut dan operasi.

Berikut merupakan *class diagram* yang terdapat pada aplikasi ini.



Gambar 5.7 Class diagram pada aplikasi

Pada gambar 5.7 terdapat *class diagram* yang terdapat pada aplikasi ini, disertai relasi dan kardinalitas antar kelas.

5.5 Pseudocode

Pseudocode merupakan sebuah bahasa penggambaran algoritme dalam penyelesaian sebuah permasalahan, yang berfungsi mempermudah programmer dalam menulis kode dan memahami algoritme dari kode tersebut tanpa harus memikirkan pemilihan bahasa pemrograman dan sintaks yang akan digunakan. Pada bagian ini akan dijabarkan beberapa *pseudocode* yang menggambarkan fungsionalitas utama sistem yang mengacu pada *sequence diagram*.

5.5.1 Pseudocode getAllRecipe()

Pseudocode getAllRecipe() menggambarkan algoritme dalam menampilkan daftar resep pada aplikasi.

Tabel 5.5 Pseudocode getAllRecipe()

Pseudocode 1: getAllRecipe()	
1	Input: -
2	Output: listRecipe
3	
4	Start
5	Initialize variable listRecipe = array
6	Get all recipe
7	When response:
8	Is success:
9	Set listRecipe = response data
10	Notify new listRecipe
11	Is error:
12	Log the error message
13	Return listRecipe
14	End

Pada tabel 5.5 terdapat *pseudocode* getAllRecipe() dengan input tidak ada dan output berupa daftar resep, yang dimulai dari mengambil semua resep, dan mengembalikan daftar resep jika respon berhasil atau melakukan logging terhadap error jika respon error.

5.5.2 Pseudocode saveRecipe(recipeId)

Pseudocode saveRecipe(recipeId) menggambarkan algoritme dalam melakukan simpan resep pada aplikasi.

Tabel 5.6 Pseudocode saveRecipe(recipeId)

Pseudocode 2: saveRecipe(recipeId)

1	Input: recipeId
2	Output: status
3	
4	Start
5	Input recipeId
6	Initialize variable status = loading
7	Save recipeId to user data
8	When response
9	Is success
10	Notify latest data
11	Set status = success
12	Notify latest status
13	Is error
14	Set status = error
15	Notify latest status
16	End

Pada tabel 5.6 terdapat *pseudocode* saveRecipe(recipeId) dengan input id resep dan output status, yang dimulai dari input id resep dan menyimpannya ke data pengguna, dan mengembalikan loading sebagai status saat masih loading, jika tidak dalam keadaan loading maka akan mengembalikan success sebagai status dan memperbarui data resep tersimpan terbaru jika respon berhasil, atau menampilkan mengembalikan error sebagai status jika respon error.

5.5.3 Pseudocode removeRecipeFromSaved(recipeId)

Pseudocode removeRecipeFromSaved(recipeId) menggambarkan algoritme dalam melakukan hapus resep tersimpan pada aplikasi.

Tabel 5.7 Pseudocode removeRecipeFromSaved(recipeId)

<i>Pseudocode</i> 3: removeRecipeFromSaved(recipeId)	
1	Input: recipeId
2	Output: status
3	
4	Start
5	Input recipeId
6	Initialize variable status = loading
7	Delete saved recipeId from user data
8	When response
9	Is success
10	Notify latest data
11	Set status = success
12	Notify latest status

13	Is error
14	Set status = error
15	Notify latest status
16	End

Pada tabel 5.7 terdapat *pseudocode* removeRecipeFromSaved(recipeId) dengan input id resep dan output status, yang dimulai dari input id resep dan menghapuskan id resep tersebut dari daftar resep tersimpan pada data pengguna, dan mengembalikan loading sebagai status saat masih loading, jika tidak dalam keadaan loading maka akan mengembalikan success sebagai status dan memperbarui data resep tersimpan terbaru jika respon berhasil, atau menampilkan mengembalikan error sebagai status jika respon error.

5.5.4 Pseudocode addPersonalNote(recipeId, note)

Pseudocode addPersonalNote(recipeId, note) menggambarkan algoritme dalam melakukan tambah catatan pribadi pada resep pada aplikasi.

Tabel 5.8 Pseudocode addPersonalNote(recipeId, note)

<i>Pseudocode</i> 4: addPersonalNote(recipeId, note)	
1	Input: recipeId, note
2	Output: status
3	
4	Start
5	Input recipeId, note
6	Initialize variable status = loading
7	Save note to user data on corresponding recipeId
8	When response:
9	Is success
10	Set status = success
11	Notify new status
12	Is error
13	Set status = error
14	Notify new status
15	Return status
16	End

Pada tabel 5.8 terdapat *pseudocode* addPersonalNote(recipeId, note) dengan input id resep serta catatan dan output status, yang dimulai dari input id resep dan catatan, menyimpan catatan tersebut ke data pengguna dengan id resep yang sesuai, mengembalikan loading sebagai status saat masih loading, jika tidak dalam keadaan loading maka akan mengembalikan success sebagai status jika

respon berhasil atau menampilkan mengembalikan error sebagai status jika respon error.

BAB 6 IMPLEMENTASI

1 6.1 Spesifikasi Sistem

Pada bagian ini akan dijabarkan spesifikasi sistem yang digunakan mengembangkan aplikasi ini. Spesifikasi sistem tersebut terdiri dari spesifikasi perangkat keras, spesifikasi perangkat keras perangkat bergerak, dan spesifikasi perangkat lunak.

6.1.1 Spesifikasi Perangkat Keras

Spesifikasi dari perangkat keras yang digunakan dalam mengembangkan aplikasi ini akan dijabarkan pada tabel 6.1.

Tabel 6.1 Spesifikasi perangkat keras

Komponen	Spesifikasi
Model	Asus F570zd
Prosesor	AMD Ryzen 5 2500u max 3.6GHz
Memori	8 GB + 8 GB DDR4 Dual Channel
Penyimpanan	1000 GB SSD + 1000 GB HDD

2 Pada tabel 6.1 terdapat spesifikasi perangkat keras yang digunakan dalam mengembangkan aplikasi yang terdiri dari Model, prosesor, memori, dan penyimpanan.

1 6.1.2 Spesifikasi Perangkat Keras Perangkat Bergerak

Spesifikasi dari perangkat keras perangkat bergerak yang digunakan dalam mengembangkan aplikasi ini akan dijabarkan pada tabel 6.2.

Tabel 6.2 spesifikasi perangkat keras perangkat bergerak

Komponen	Spesifikasi
Model	Samsung A52s 5G
Prosesor	Qualcomm Snapdragon 778G max 2.4GHz
Memori	8 GB
Penyimpanan	256 GB

1 Pada tabel 6.2 terdapat spesifikasi perangkat keras perangkat bergerak yang digunakan dalam mengembangkan aplikasi yang terdiri dari model, prosesor, memori, dan penyimpanan.

1 6.1.3 Spesifikasi Perangkat Lunak

Spesifikasi dari perangkat lunak yang digunakan dalam mengembangkan aplikasi ini akan dijabarkan pada tabel 6.1.

Tabel 6.3 Spesifikasi perangkat lunak

Komponen	Spesifikasi
Sistem operasi	Windows 10
Bahasa pemrograman	Kotlin
Alat pengembangan	Android Studio Chipmunk 2021.2.1 Patch 1

1
Pada tabel 6.1 terdapat spesifikasi perangkat lunak yang digunakan dalam mengembangkan aplikasi yang terdiri dari sistem operasi, bahasa pemrograman, dan alat pengembangan.

1 6.2 Implementasi Database

Pada bagian ini akan dijabarkan hasil implementasi dari perancangan database yang terdapat pada bab sebelumnya. Entitas *database* yang terdapat pada pengembangan perangkat lunak ini yaitu MahasiswaKos, Recipe, PersonalNote, dan Review.

6.2.1 Implementasi Database Entitas MahasiswaKos

Implementasi dari entitas MahasiswaKos yang mengacu pada tabel 5.1 dapat dilihat pada gambar 6.1.

The screenshot shows the Firebase Cloud Firestore interface. At the top, there's a navigation bar with 'mahasiswa_kos' selected. Below it, there's a list of collections: 'mahasiswa_kos' (selected), 'PersonalNote', 'Review', and 'User'. Under 'mahasiswa_kos', there's a single document named 'mahasiswa_kos'. This document has three fields: 'recipies' (a list of recipe documents), 'name' (with the value 'Muhammad Ziyad Al Mawardi'), and 'personalNote' (with the value 'penyelidikan rasa digeluti kafolu jumur lebet enak').

Gambar 6.1 Implementasi Entitas MahasiswaKos

Pada gambar 6.1 terdapat entitas MahasiswaKos yang telah diimplementasikan dalam bentuk database pada Firebase Cloud Firestore dengan nama "mahasiswa_kos".

6.2.2 Implementasi Database Entitas Recipe

Implementasi dari entitas Recipe yang mengacu pada tabel 5.2 dapat dilihat pada gambar 6.2.

Gambar 6.2 Implementasi Entitas Recipe

Pada gambar 6.2 terdapat entitas Recipe yang telah diimplementasikan dalam bentuk database pada Firebase Cloud Firestore dengan nama recipes.

6.2.3 Implementasi Database Entitas PersonalNote

Implementasi dari entitas PersonalNote yang mengacu pada tabel 5.3 dapat dilihat pada gambar 6.3.

Gambar 6.3 Implementasi Entitas PersonalNote

Pada gambar 6.3 terdapat entitas PersonalNote yang telah diimplementasikan dalam bentuk database pada Firebase Cloud Firestore. Entitas PersonalNote terdapat pada entitas MahasiswaKos pada variabel array bernama "listPersonalNote".

6.2.4 Implementasi Database Entitas Review

Implementasi dari entitas Review yang mengacu pada tabel 5.4 dapat dilihat pada gambar 6.4.



Gambar 6.4 Implementasi Entitas Review

Pada gambar 6.4 terdapat entitas *Review* yang telah diimplementasikan dalam bentuk database pada Firebase Cloud Firestore. Entitas *Review* terdapat pada entitas *Recipe* pada variabel *array* bernama *listReview*.

1

6.3 Implementasi Kode Program

Pada bagian ini akan dijabarkan hasil implementasi dari perancangan *pseudocode* yang terdapat pada bab sebelumnya.

6.3.1 Implementasi Kode Program getAllRecipe()

Implementasi kode program *getAllRecipe()* menggambarkan algoritme secara implementatif dalam menampilkan daftar resep pada aplikasi. Implementasi kode program *getAllRecipe()* pada *pseudocode* di tabel 5.5.

Tabel 6.4 Implementasi kode getAllRecipe()

Kode program 1: getAllRecipe()	
1	<pre> 1 fun getAllRecipe(): MutableLiveData<ArrayList<Recipe>> { 2 database.collection("recipes") 3 .addSnapshotListener { snapshot, e -> 4 if (e != null) { 5 Log.w("TEZ", "Listen failed.", e) 6 return@addSnapshotListener 7 } 8 9 if (snapshot != null && !snapshot.isEmpty) { 10 val allRecipe = arrayListOf<Recipe>() 11 for (doc in snapshot) { 12 allRecipe.add(doc.toObject(Recipe::class.java)) 13 Log.d("TEZ", "Current data: \$doc") 14 } 15 allRecipeLiveData.postValue(allRecipe) 16 } else { 17 Log.d("TEZ", "Current data: null") 18 } 19 } 20 } 21 }</pre>

```

18     }
19 }
20
21     return allRecipeLiveData
22 }

```

Pada tabel 6.4 terdapat implementasi kode program getAllRecipe(). Pada baris 1 terdapat nama fungsi beserta parameter dan nilai kembalinya. Pada baris 2 hingga 22 akan dilakukan pengambilan terhadap data semua resep. Pada baris 4 hingga 7 terdapat percabangan apabila response terhadap permintaan ke data gagal, lalu akan melakukan *log* terhadap pesan *error* tersebut. Pada baris 9 hingga 15 terdapat percabangan apabila response terhadap permintaan ke data berhasil, lalu akan melakukan pembaharuan dan *notify* terhadap daftar semua resep terbaru. Pada baris 16 hingga 17 terdapat percabangan apabila data hasil response kosong, lalu akan melakukan *log* terhadap kekosongan data tersebut.

2

6.3.2 Implementasi Kode Program saveRecipe(recipeId)

Implementasi kode program saveRecipe(recipeId) menggambarkan algoritme secara implementatif dalam melakukan simpan resep pada aplikasi. Implementasi kode program saveRecipe(recipeId) pada pseudocode di tabel 5.6.

Tabel 6.5 Implementasi kode program saveRecipe(recipeId)

Kode program 2: saveRecipe(recipeId)

```

1 suspend fun saveRecipe(recipeId: String):
2     MutableLiveData<String> {
3         val currentUser = auth.currentUser
4         val statusLiveData = MutableLiveData<String>()
5         statusLiveData.postValue("LOADING")
6         currentUser?.let {
7             database.collection("mahasiswa_kos")
8                 .document(it.uid)
9                 .get()
10                .addOnSuccessListener { document ->
11                    if (document != null) {
12                        val mahasiswaKos = document
13                            .toObject(MahasiswaKos::class.java)
14                            Log.d(TAG,
15                                "DocumentSnapshot data: ${document.data}")
16                        mahasiswaKos?.listSavedRecipeId?
17                            .add(recipeId)
18                        database.collection("mahasiswa_kos")
19                            .document(currentUser!!.uid)

```

```

20          .set(mahasiswaKos!!)
21          .addOnSuccessListener {
22              Log.d("TEZ", "Current data: $it")
23              savedRecipeId.postValue(
24                  mahasiswaKos.listSavedRecipeId)
25              statusLiveData.postValue("SUCCESS")
26          }.addOnFailureListener {
27              Log.d("TEZ", "Current data: $it")
28              statusLiveData
29                  .postValue(it.toString())
30          }
31      } else {
32          Log.d(TAG, "No such document")
33          statusLiveData.postValue("No such document")
34      }
35  }
36  .addOnFailureListener { exception ->
37      Log.d(TAG, "get failed with ", exception)
38      statusLiveData.postValue(exception.toString())
39  }
40  }
41  return statusLiveData
42 }

```

Pada tabel 6.5 terdapat implementasi kode program saveRecipe(recipeld). Pada baris 1 dan 2 terdapat nama fungsi beserta parameternya dan nilai kembalinya. Pada baris 3 diinisiasi variabel baru terkait data pengguna saat ini. Pada baris 4 dan 5 diinisiasi variabel baru untuk menyimpan data status dan memberikan loading sebagai nilai awal. Pada baris 6 hingga 42 akan dilakukan penambahan recipeld ke daftar recipeld tersimpan yaitu listRecipeld pada data entitas pengguna terkait. Pada baris 21 hingga 25 terdapat percabangan apabila response terhadap permintaan penyimpanan ke data berhasil, lalu akan melakukan pembaharuan dan *notify* terhadap daftar recipeld terbaru, kemudian akan melakukan pembaharuan dan *notify* terhadap status terbaru dengan nilai *success*. Pada baris 26 hingga 39 terdapat beberapa percabangan apabila response terhadap permintaan penyimpanan ke data gagal, lalu akan melakukan pembaharuan dan *notify* terhadap status terbaru dengan nilai *error*. Pada baris 41 akan mengembalikan nilai status.

6.3.3 Implementasi Kode Program removeRecipeFromSaved(recipeld)

Implementasi kode program removeRecipeFromSaved(recipeld) menggambarkan algoritme secara implementatif dalam melakukan hapus resep

tersimpan pada aplikasi. Implementasi kode program removeRecipeFromSaved(recipeId) pada pseudocode di tabel 5.7.

Tabel 6.6 Implementasi kode program removeRecipeFromSaved(recipeId)

Kode program 3: removeRecipeFromSaved(recipeId)	
1	suspend fun removeRecipeFromSaved(recipeId: String): 2 MutableLiveData<String> { 3 val currentUser = auth.currentUser 4 val statusLiveData = MutableLiveData<String>() 5 statusLiveData.postValue("LOADING") 6 currentUser?.let { 7 database.collection("mahasiswa_kos") 8 .document(it.uid) 9 .get() 10 .addOnSuccessListener { document -> 11 if (document != null) { 12 val mahasiswaKos = document 13 .toObject(MahasiswaKos::class.java) 14 Log.d(TAG, 15 "DocumentSnapshot data: \${document.data}") 16 mahasiswaKos?.listSavedRecipeId? 17 .remove(recipeId) 18 database.collection("mahasiswa_kos") 19 .document(currentUser!!.uid) 20 .set(mahasiswaKos!!) 21 .addOnSuccessListener { 22 Log.d("TEZ", "Current data: \$it") 23 savedRecipeId.postValue(24 mahasiswaKos.listSavedRecipeId) 25 statusLiveData.postValue("SUCCESS") 26 }.addOnFailureListener { 27 Log.d("TEZ", "Current data: \$it") 28 statusLiveData.postValue(29 it.toString()) 30 } 31 } else { 32 Log.d(TAG, "No such document") 33 statusLiveData.postValue("No such document") 34 } 35 } 36 .addOnFailureListener { exception ->

37	Log.d(TAG, "get failed with ", exception)
38	statusLiveData.postValue(exception.toString())
39	}
40	}
41	return statusLiveData
42	}

Pada tabel 6.6 terdapat implementasi kode program removeRecipeFromSaved(recipeId). Pada baris 1 dan 2 terdapat nama fungsi beserta parameternya dan nilai kembalinya. Pada baris 3 diinisiasi variabel baru terkait data pengguna saat ini. Pada baris 4 dan 5 diinisiasi variabel baru untuk menyimpan data status dan memberikan loading sebagai nilai awal. Pada baris 6 hingga 42 akan dilakukan penghapusan recipeId dari daftar recipeId tersimpan yaitu listRecipeId pada data entitas pengguna terkait. Pada baris 21 hingga 25 terdapat percabangan apabila response terhadap permintaan penyimpanan ke data berhasil, lalu akan melakukan pembaharuan dan *notify* terhadap daftar recipeId terbaru, kemudian akan melakukan pembaharuan dan *notify* terhadap status terbaru dengan nilai *success*. Pada baris 26 hingga 39 terdapat beberapa percabangan apabila response terhadap permintaan penyimpanan ke data gagal, lalu akan melakukan pembaharuan dan *notify* terhadap status terbaru dengan nilai *error*. Pada baris 41 akan mengembalikan nilai status.

6.3.4 Implementasi Kode Program addPersonalNote(recipeId, note)

Implementasi kode program addPersonalNote(recipeId, note) menggambarkan algoritme secara implementatif dalam melakukan tambah catatan pribadi pada resep pada aplikasi. Implementasi kode program addPersonalNote(recipeId, note) pada pseudocode di tabel 5.8.

Tabel 6.7 Implementasi kode program addPersonalNote(recipeId, note)

Kode program 4: addPersonalNote(recipeId, note)	
1	suspend fun addPersonalNote(recipeId: String, note: String):
2	MutableLiveData<String> {
3	val currentUser = auth.currentUser
4	val statusLiveData = MutableLiveData<String>()
5	statusLiveData.postValue("LOADING")
6	currentUser?.let {
7	database.collection("mahasiswa_kos")
8	.document(currentUser.uid).get()
9	.addOnSuccessListener { snapshot ->
10	if (snapshot != null && snapshot.exists()) {
11	val mPersonalNote = PersonalNote(recipeId,
12	note)

```

13         val mMahasiswaKos = snapshot
14             .toObject(MahasiswaKos::class.java)!!
15             Log.d("TEZZ", "Current data: $mMahasiswaKos")
16             for (i in mMahasiswaKos.listPersonalNote) {
17                 if (i.recipeId == recipeId) {
18                     mMahasiswaKos.listPersonalNote
19                         .remove(i)
20                 }
21             }
22             mMahasiswaKos.listPersonalNote
23                 .add(mPersonalNote)
24                 database.collection("mahasiswa_kos")
25                     .document(currentUser.uid).set(mMahasiswaKos)
26                     .addOnSuccessListener {
27                         statusLiveData.postValue("SUCCESS")
28                         Log.d("TEZZ", "SUCCESS")
29                     }.addOnFailureListener {
30                         statusLiveData
31                             .postValue(it.toString())
32                     }
33             } else {
34                 Log.d("TEZ", "Current data: null")
35                 statusLiveData.postValue("Null")
36             }
37         }
38     }
39     return statusLiveData
40 }
```

Pada tabel 6.7 terdapat implementasi kode program addPersonalNote(recipeId, note). Pada baris 1 dan 2 terdapat nama fungsi beserta parameter dan nilai kembalinya. Pada baris 3 diinisiasi variabel baru terkait data pengguna saat ini. Pada baris 4 dan 5 diinisiasi variabel baru untuk menyimpan data status dan memberikan loading sebagai nilai awal. Pada baris 6 hingga 38 akan dilakukan penyimpanan catatan pribadi terhadap sebuah resep. Pada baris 26 hingga 28 terdapat percabangan apabila response terhadap permintaan penyimpanan ke data berhasil, lalu akan melakukan pembaharuan dan *notify* terhadap status terbaru dengan nilai *success*. Pada baris 29 hingga 36 terdapat percabangan apabila response terhadap permintaan penyimpanan ke data gagal, lalu akan melakukan pembaharuan dan *notify* terhadap status terbaru dengan nilai *error*. Pada baris 40 akan mengembalikan nilai status.

1

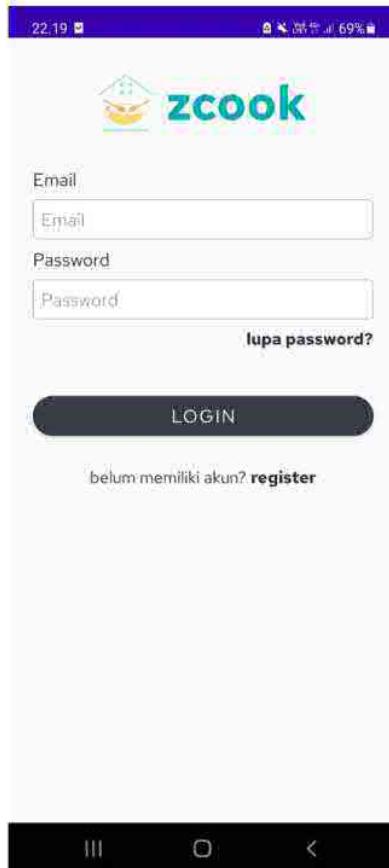
6.4 Implementasi Antarmuka Pengguna

Pada bagian ini akan dijabarkan hasil implementasi dari perancangan antarmuka pengguna yang terdapat pada bab sebelumnya.

1

6.4.1 Implementasi Halaman Login

Berikut merupakan implementasi halaman *login*.



1

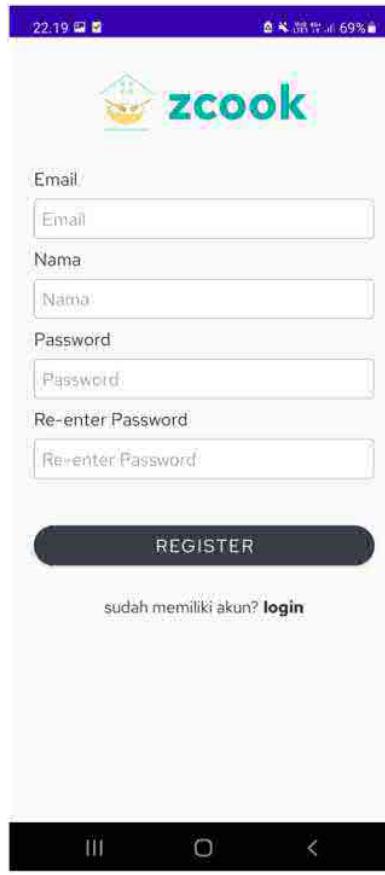
Gambar 6.5 Implementasi halaman *login*

Pada halaman *login*, terdapat kolom untuk mengisi *email* dan *password* yang nantinya akan diisi oleh pengguna, dan divalidasi oleh sistem saat tombol *login* ditekan. Terdapat pula teks *register* yang menavigasikan pengguna ke halaman *register*, dan teks *lupa password* yang menavigasikan pengguna ke halaman *reset password*.

2

6.4.2 Implementasi Halaman Register

Berikut merupakan implementasi halaman *register*.



Gambar 6.6 Implementasi halaman *register*

Pada halaman *register*, terdapat kolom untuk mengisi *email*, nama, *password*, dan *re-enter password* yang nantinya akan diisi oleh pengguna, dan divalidasi oleh sistem saat tombol *register* ditekan. *Password* dan *re-enter password* akan dicek kesamaannya, dan *email* akan dicek apakah masih tersedia dan belum terpakai oleh akun lain oleh sistem. Terdapat pula teks *login* yang menavigasikan pengguna ke halaman *login*.

6.4.3 Implementasi Halaman Reset Password

Berikut merupakan implementasi halaman *register*.



Gambar 6.7 Implementasi halaman reset password

Pada halaman *reset password*, terdapat kolom untuk mengisi *email* yang nantinya akan diisi oleh pengguna, dan menjadi tujuan pengiriman *link* *reset password* saat tombol *kirim* ditekan. Terdapat pula teks *register* yang menavigasikan pengguna ke halaman *register*, dan tombol *back* untuk kembali ke halaman sebelumnya.

6.4.4 Implementasi Halaman Daftar Resep

Berikut merupakan implementasi halaman daftar resep.

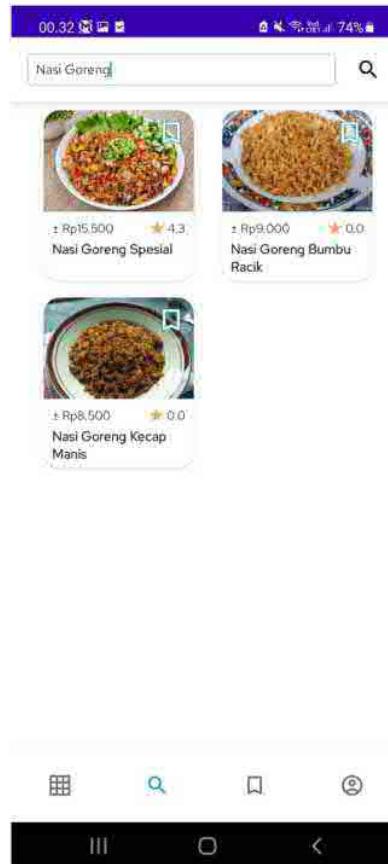


Gambar 6.8 Implementasi halaman daftar resep

Pada halaman daftar resep, terdapat rekomendasi resep sesuai harganya, dalam hal ini resep 10ribuan dan resep di bawah 10ribu, dan juga terdapat daftar semua resep yang tersedia. Tiap resep yang ditampilkan terdiri atas nama resep, gambar masakan, icon simpan resep, *rating*, dan estimasi harga pembuatan masakan. Jika pengguna belum masuk, maka icon simpan resep saat diklik akan navigasikan pengguna ke halaman *login*, dan jika sudah masuk maka icon simpan resep saat diklik akan menyimpan resep tersebut ke daftar resep. Resep yang ditekan akan navigasikan pengguna ke halaman detail dari resep tersebut. Terdapat juga tombol pencarian yang akan navigasikan pengguna ke halaman pencarian, dan *bottom navigation bar* atau bilah navigasi bawah yang terdiri dari menu daftar resep, resep tersimpan, dan pengaturan yang akan navigasikan masing-masing ke halaman daftar resep, halaman resep tersimpan, dan halaman pengaturan.

6.4.5 Implementasi Halaman Pencarian Resep

Berikut merupakan implementasi halaman pencarian resep.



Gambar 6.9 Implementasi halaman pencarian resep

Pada halaman pencarian resep, terdapat kolom untuk mengisi kata kunci pencarian, tombol cari untuk melakukan pencarian, dan tombol back untuk kembali ke halaman sebelumnya. Terdapat pula daftar hasil pencarian resep. Tiap resep yang ditampilkan terdiri atas nama resep, gambar masakan, icon simpan resep, rating, dan estimasi harga pembuatan masakan. Jika pengguna belum masuk, maka icon simpan resep saat diklik akan navigasikan pengguna ke halaman login, dan jika sudah masuk maka icon simpan resep saat diklik akan menyimpan resep tersebut ke daftar resep. Resep yang ditekan akan navigasikan pengguna ke halaman detail dari resep tersebut.

6.4.6 Implementasi Halaman Detail Resep Tamu

Berikut merupakan implementasi halaman login.

Nasi Goreng Spesial

★ 4,3.

Total harga : ± Rp15.500

Estimasi waktu memasak : ± 20 menit

Bahan :

-Nasi putih 1 piring	± Rp 3.000
-Bawang putih 2 siung	± Rp 500
-Kecap manis 1 sdm	± Rp 1.000 (20ml)
-Saus sambal 1 sdm	± Rp 500 (10gr)
-Saus tiram 1 sdm	± Rp 3.000 (23ml)
-Garam 1 sdt	± Rp 2.000 (200gr)
-Penyedap rasa 1 sdt	± Rp 500 (8gr)
-Daun bawang 1 batang	± Rp 1.000 (80gr)
-Telur ayam 1 butir	± Rp 2.000
-Sosis ayam 1 buah, iris	± Rp 1.000
-Minyak goreng 3 sdm	± Rp 1000 (3 sdm)

Cara membuat :

- Siapkan penggorengan dengan api sedang, tuang margarin atau minyak goreng.
- Masukkan bawang putih dan daun bawang yang sudah dicincang halus. Tumis hingga berbau harum atau hingga warnanya keemasan.
- Masukkan sosis dan 1 butir telur ayam. Tumis sebentar.
- Masukkan bumbu halus dan nasi. Aduk hingga tercampur rata.
- Tuang kecap manis, saus sambal, saus tiram, garam, dan kaldu bubuk. Aduk hingga warna nasi berubah secara merata.
- Nasi goreng biasa yang sederhana, dan enak siap disajikan.

Cara membuat :

1. Siapkan penggorengan dengan api sedang,

Kata orang-orang tentang resep ini

Muhammad Ziyad Al Maududi
enak

Ihsan
enak banget

Aldi

Gambar 6.10 Implementasi halaman detail resep Tamu

Pada halaman detail resep, terdapat nama resep, gambar masakan, bahan-bahan yang dibutuhkan, harga dari tiap bahan, harga total dari resep tersebut, estimasi waktu yang dibutuhkan, rating dan review, dan langkah-langkah memasaknya. Terdapat juga tombol kembali untuk kembali ke halaman sebelumnya, icon simpan resep, dan tombol masuk ke akun yang akan menavigasikan pengguna ke halaman *login*. Jika diklik, icon simpan resep akan menavigasikan pengguna ke halaman *login*.

6.4.7 Implementasi Halaman Detail Resep Mahasiswa Kos

Berikut merupakan implementasi halaman *login*.



Nasi Goreng Spesial

4.3

Total harga : ± Rp15.500

Estimasi waktu memasak : ± 20 menit

Bahan :

-Nasi putih 1 piring	± Rp 3.000
-Bawang putih 2 siung	± Rp 500
-Kecap manis 1 sdm	± Rp 1.000 (20ml)
-Saus sambal 1 sdm	± Rp 500 (10gr)
-Saus tiram 1 sdm	± Rp 3.000 (23ml)
-Garam 1 sdt	± Rp 2.000 (200gr)
-Penyedap rasa 1 sdt	± Rp 500 (8gr)
-Daun bawang 1 batang	± Rp 1.000 (80gr)
-Telur ayam 1 butir	± Rp 2.000
-Sosis ayam 1 buah, iris	± Rp 1.000
-Minyak goreng 3 sdm	± Rp 1000 (3 sdm)

Cara membuat:

1. Siapkan pengorengan dengan api sedang,

Cara membuat :

- Siapkan pengorengan dengan api sedang, tuangi margarin atau minyak goreng.
- Masukkan bawang putih dan daun bawang yang sudah dicincang halus. Tumis hingga berbau harum atau hingga warnanya keemasan.
- Masukkan sosis dan 1 butir telur ayam. Tumis sebentar.
- Masukkan bumbu halus dan nasi. Aduk hingga tercampur rata.
- Tuang kecap manis, saus sambal, saus tiram, garam, dan kaldu bubuk. Aduk hingga warna nasi berubah secara merata.
- Nasi goreng biasa yang sederhana, dan enak, siap disajikan,

Catatan pribadi

penyedap rasa diganti kaldu jamur lebih enak

SIMPAN CATATAN

Review resep ini!



enak

SIMPAN REVIEW



Gambar 6.11 Implementasi halaman detail resep Mahasiswa Kos

Pada halaman detail resep, terdapat nama resep, gambar masakan, bahan-bahan yang dibutuhkan, harga dari tiap bahan, harga total dari resep tersebut, estimasi waktu yang dibutuhkan, *rating* dan *review*, dan langkah-langkah memasaknya. Terdapat kolom untuk mengisi catatan pribadi dan tombol untuk menyimpan catatan, bintang untuk memberikan *rating* beserta kolom untuk mengisi *review* dan tombol untuk menyimpan *review*, icon simpan resep, dan tombol kembali untuk kembali ke halaman sebelumnya. Jika diklik, icon simpan resep akan menyimpan resep tersebut ke daftar resep.

6.4.8 Implementasi Halaman Resep Tersimpan Tamu

Berikut merupakan implementasi halaman *login*.

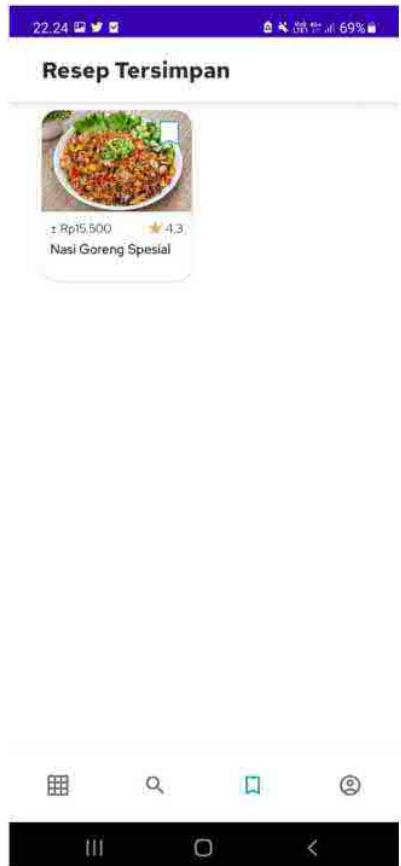


Gambar 6.12 Implementasi halaman resep tersimpan Tamu

Pada halaman resep tersimpan, terdapat tombol masuk ke akun yang akan menavigasikan pengguna ke halaman *login*. Terdapat juga *bottom navigation bar* atau bilah navigasi bawah yang terdiri dari menu daftar resep, resep tersimpan, dan pengaturan yang akan menavigasikan masing-masing ke halaman daftar resep, halaman resep tersimpan, dan halaman pengaturan.

6.4.9 Implementasi Halaman Resep Tersimpan Mahasiswa Kos

Berikut merupakan implementasi halaman *login*.



Gambar 6.13 Implementasi halaman resep tersimpan Mahasiswa Kos

Pada halaman resep tersimpan, terdapat daftar resep yang telah kita simpan. Tiap resep yang ditampilkan terdiri atas nama resep, gambar masakan, icon simpan resep, rating, dan estimasi harga pembuatan masakan. Jika icon simpan resep diklik, maka resep tersebut akan dihapus dari daftar resep tersimpan. Resep yang ditekan akan menavigasikan pengguna ke halaman detail dari resep tersebut. Terdapat juga *bottom navigation bar* atau bilah navigasi bawah yang terdiri dari menu daftar resep, resep tersimpan, dan pengaturan yang akan menavigasikan masing-masing ke halaman daftar resep, halaman resep tersimpan, dan halaman pengaturan.

6.4.10 Implementasi Halaman Pengaturan Tamu

Berikut merupakan implementasi halaman pengaturan tamu.

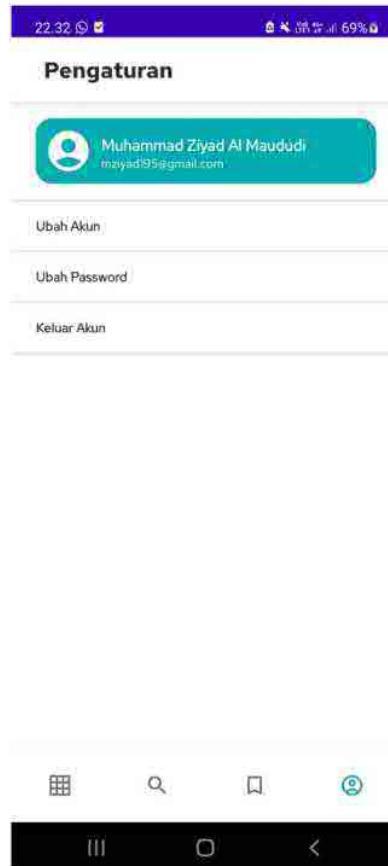


Gambar 6.14 Implementasi halaman pengaturan Tamu

Pada halaman pengaturan Tamu, terdapat tombol masuk ke akun yang akan menavigasikan pengguna ke halaman *login*. Terdapat juga *bottom navigation bar* atau bilah navigasi bawah yang terdiri dari menu daftar resep, resep tersimpan, dan pengaturan yang akan menavigasikan masing-masing ke halaman daftar resep, halaman resep tersimpan, dan halaman pengaturan.

6.4.11 Implementasi Halaman Pengaturan Mahasiswa Kos

Berikut merupakan implementasi halaman pengaturan Mahasiswa Kos.



Gambar 6.15 Implementasi halaman pengaturan Mahasiswa Kos

Pada halaman pengaturan, terdapat nama pengguna, *email* pengguna, tombol ubah akun, tombol ubah *password*, dan tombol keluar akun. Tombol ubah akun akan navigasikan pengguna ke halaman ubah akun, tombol ubah *password* akan navigasikan pengguna ke halaman ubah *password*, dan tombol keluar akun akan mengeluarkan pengguna dari hak akses pengguna Mahasiswa Kos dan navigasikan pengguna ke halaman *login*. Terdapat juga *bottom navigation bar* atau bilah navigasi bawah yang terdiri dari menu daftar resep, resep tersimpan, dan pengaturan yang akan navigasikan masing-masing ke halaman daftar resep, halaman resep tersimpan, dan halaman pengaturan.

6.4.12 Implementasi Halaman Ubah Akun

Berikut merupakan implementasi halaman ubah akun.

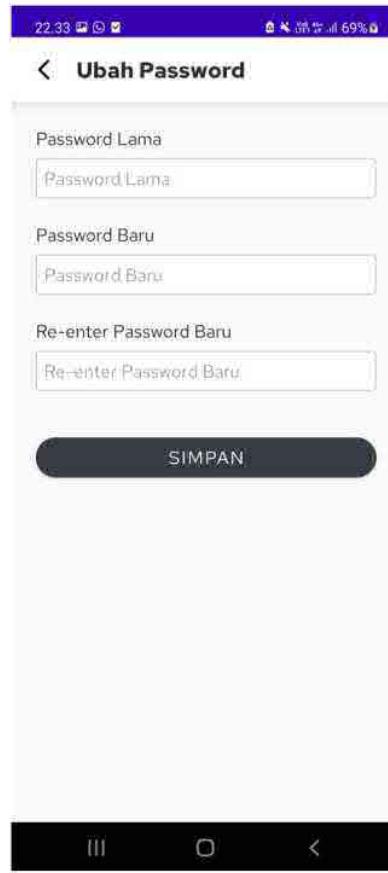


Gambar 6.16 Implementasi halaman ubah akun Mahasiswa Kos

Pada halaman ubah akun, terdapat kolom untuk mengisi nama, dan email, yang nantinya akan diisi oleh pengguna, dan diproses untuk disimpan oleh sistem saat tombol simpan ditekan. Terdapat juga tombol kembali, untuk kembali ke halaman sebelumnya.

6.4.13 Implementasi Halaman Ubah Password

Berikut merupakan implementasi halaman ubah password.



Gambar 6.17 Implementasi halaman ubah **password** Mahasiswa Kos

Pada halaman ubah **password**, terdapat kolom untuk mengisi **password lama**, **password baru**, dan **re-enter password baru**, yang nantinya akan diisi oleh pengguna, dan diproses untuk disimpan oleh sistem saat tombol simpan ditekan. **Password lama** akan dicek oleh sistem validitasnya, dan **password** dan **re-enter password** akan dicek kesamaannya. Terdapat juga tombol kembali, untuk kembali ke halaman sebelumnya.

BAB 7 PENGUJIAN

2 7.1 Black Box Testing

Black box testing merupakan sebuah pengujian yang dilakukan dengan membuat kasus uji terhadap yang mengacu kepada setiap fitur, dan akan diuji oleh pengembang. Pada bagian ini akan dijabarkan hasil *black box testing* pada aplikasi ini. Daftar pengujian yang dilakukan pada *black box testing* mengacu pada *user scenario* yang terdapat pada tabel 4.7, dan fitur-fitur pada subbab 4.3.

7.1.1 Black Box Testing Fitur Register

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 1, dan fitur pada gambar 4.14.

Tabel 7.1 Black box testing fitur register

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	<i>Register</i> dengan benar	Pengujian dengan Tamu mengisi nama, <i>email</i> yang belum pernah didaftarkan, dan <i>password</i> dan re-enter <i>password</i> dengan isi yang sama	Tamu terdaftar dan masuk ke sistem sebagai MahasiswaKos	Berhasil
2.	<i>Register</i> dengan mengosongkan salah satu kolom	Pengujian dengan Tamu mengosongkan salah satu kolom, baik itu nama, <i>email</i> , <i>password</i> , maupun re-enter <i>password</i>	Tamu gagal mendaftar dan muncul pesan "Lengkapi data"	Berhasil
3.	<i>Register</i> dengan <i>email</i> yang sudah terdaftar	Pengujian dengan Tamu mengisi nama, <i>password</i> dan re-	Tamu gagal mendaftar dan muncul pesan kegagalan	Berhasil

		enter <i>password</i> dengan isi yang sama, namun dengan <i>email</i> yang sudah pernah didaftarkan		
4.	<i>Register</i> dengan <i>password</i> dan re-enter <i>password</i> yang berbeda	Pengujian dengan Tamu mengisi nama, <i>email</i> yang belum pernah didaftarkan, namun <i>password</i> dan re-enter <i>password</i> dengan isi yang berbeda	Tamu gagal mendaftar dan muncul pesan "Password dan Re-enter Password tidak cocok"	Berhasil

Pada tabel 7.1 terdapat nama beserta deskripsi pengujian terhadap fitur *register*.

7.1.2 Black Box Testing Fitur Login

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 2, dan fitur pada gambar 4.15.

Tabel 7.2 Black box testing fitur login

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	<i>Login</i> dengan benar	Pengujian dengan Tamu mengisi nama, <i>email</i> yang pernah didaftarkan, dan <i>password</i> yang sesuai	Tamu masuk ke sistem sebagai MahasiswaKos	Berhasil
2.	<i>Login</i> dengan mengosongkan	Pengujian dengan Tamu	Tamu gagal masuk ke sistem	Berhasil

	salah satu kolom	mengosongkan salah satu kolom, baik itu <i>email</i> , maupun <i>password</i>	dan muncul pesan "Lengkapi data"	
3.	<i>Login</i> dengan <i>email</i> dan <i>password</i> yang tidak sesuai	Pengujian dengan Tamu mengisi <i>email</i> dan <i>password</i> yang tidak sesuai	Tamu gagal masuk ke sistem dan muncul pesan kegagalan	Berhasil

Pada tabel 7.2 terdapat nama beserta deskripsi pengujian terhadap fitur *login*.

7.1.3 Black Box Testing Fitur Ganti Password

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 3, dan fitur pada gambar 4.16.

Tabel 7.3 Black box testing fitur ganti password

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Ganti <i>password</i> dengan benar	Pengujian dengan MahasiswaKos mengisi <i>password</i> lama dengan benar, dan <i>password</i> dan re-enter <i>password</i> dengan isi yang sama	<i>Password</i> berhasil terganti dan kembali ke halaman pengaturan	Berhasil
2.	Ganti <i>password</i> dengan mengosongkan salah satu kolom	Pengujian dengan MahasiswaKos mengosongkan salah satu kolom, baik itu <i>password</i> lama, <i>password</i> ,	<i>Password</i> gagal terganti dan muncul pesan "Lengkapi data"	Berhasil

		maupun re-enter password		
3.	Ganti password dengan password lama yang salah	Pengujian dengan MahasiswaKos mengisi nama, password dan re-enter password dengan isi yang sama, namun dengan password lama yang tidak sesuai	Password gagal tergantikan dan muncul pesan kegagalan	Berhasil
4.	Ganti password dengan password dan re-enter password yang berbeda	Pengujian dengan MahasiswaKos mengisi password lama dengan benar, namun password dan re-enter password dengan isi yang berbeda	Password gagal tergantikan dan muncul pesan "Password dan Re-enter Password tidak cocok"	Berhasil

Pada tabel 7.3 terdapat nama beserta deskripsi pengujian terhadap fitur ganti password.

7.1.4 Black Box Testing Fitur Ubah Profil

Berikut merupakan hasil black box testing yang mengacu pada user scenario yang terdapat pada tabel 4.7 nomor 4, dan fitur pada gambar 4.17.

Tabel 7.4 Black box testing fitur ubah profil

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Ubah profil dengan benar	Pengujian dengan MahasiswaKos mengisi nama	Nama dan Email terbaru telah terubah dan kembali ke	Berhasil

		baru, dengan <i>email</i> baru yang belum terdaftar dalam sistem	halaman pengaturan	
2.	Ubah profil dengan mengosongkan salah satu kolom	Pengujian dengan MahasiswaKos mengosongkan salah satu kolom, baik itu <i>password lama</i> , <i>password</i> , maupun re-enter <i>password</i>	<i>Password</i> gagal terganti dan muncul pesan "Lengkapi data"	Berhasil
3.	Ubah profil dengan <i>email</i> yang sudah terdaftar	Pengujian dengan MahasiswaKos mengisi nama, <i>password</i> dan re-enter <i>password</i> dengan isi yang sama, namun dengan <i>password lama</i> yang tidak sesuai	<i>Password</i> gagal terganti dan muncul pesan kegagalan	Berhasil

Pada tabel 7.4 terdapat nama beserta deskripsi pengujian terhadap fitur ubah *password*.

7.1.5 Black Box Testing Fitur Reset Password

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 5, dan fitur pada gambar 4.18.

Tabel 7.5 Black box testing fitur reset password

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Reset <i>password</i> dengan benar	Pengujian dengan Tamu mengisi <i>email</i> untuk dikirimkan	Tautan reset <i>password</i> terkirim ke <i>email</i> , dan	Berhasil

		tautan reset password ke email tersebut	kembali ke halaman login	
2.	Reset password dengan mengosongkan kolom email	Pengujian dengan MahasiswaKos mengosongkan kolom email	Reset password gagal terganti dan muncul pesan "Masukkan email yang benar"	Berhasil

Pada tabel 7.3 terdapat nama beserta deskripsi pengujian terhadap fitur reset password.

7.1.6 Black Box Testing Fitur Menampilkan Daftar Resep

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 6, dan fitur pada gambar 4.19.

Tabel 7.6 Black box testing fitur menampilkan daftar resep

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Menampilkan daftar resep	Pengujian dengan pengguna membuka halaman daftar resep	Daftar resep ditampilkan	Berhasil

Pada tabel 7.6 terdapat nama beserta deskripsi pengujian terhadap fitur menampilkan daftar resep.

7.1.7 Black Box Testing Fitur Pencarian Resep

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 7, dan fitur pada gambar 4.20.

Tabel 7.7 Black box testing fitur pencarian resep

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Pencarian resep	Pengujian	Daftar resep	Berhasil

	ditemukan	dengan pengguna mengisi kolom pencarian dengan kata kunci yang benar dengan yang ada pada sistem dan menklik tombol cari	yang dicari ditampilkan	
2.	Pencarian resep tidak ditemukan	Pengujian dengan pengguna mengisi kolom pencarian dengan kata kunci yang tidak ada pada sistem dan menklik tombol cari	Daftar resep yang dicari kosong dan muncul pesan "Resep tidak ditemukan"	Berhasil

Pada tabel 7.7 terdapat nama beserta deskripsi pengujian terhadap fitur pencarian resep.

7.1.8 Black Box Testing Fitur Menampilkan Rekomendasi Resep Berdasarkan Harga

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 8, dan fitur pada gambar 4.21.

Tabel 7.8 Black box testing fitur menampilkan rekomendasi resep berdasarkan harga

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Menampilkan rekomendasi resep masakan berdasarkan harga	Pengujian dengan pengguna membuka halaman daftar resep	Daftar resep masakan berdasarkan harga ditampilkan	Berhasil

Pada tabel 7.8 terdapat nama beserta deskripsi pengujian terhadap fitur menampilkan rekomendasi resep berdasarkan harga.

7.1.9 Black Box Testing Fitur Menampilkan Detail Resep

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 9, dan fitur pada gambar 4.22.

Tabel 7.9 Black box testing fitur menampilkan detail resep

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Menampilkan detail resep	Pengujian dengan pengguna membuka salah satu resep	Detail resep masakan ditampilkan	Berhasil

Pada tabel 7.9 terdapat nama beserta deskripsi pengujian terhadap fitur menampilkan detail resep.

7.1.10 Black Box Testing Fitur Menambahkan Rating dan Review

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 10, dan fitur pada gambar 4.23.

Tabel 7.10 Black box testing fitur menambahkan rating dan review

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Menambahkan rating dan review	Pengujian dengan MahasiswaKos membuka salah satu resep dan menambahkan rating dan review	Rating dan review tersebut ditambahkan dan ditampilkan di resep tersebut	Berhasil

Pada tabel 7.10 terdapat nama beserta deskripsi pengujian terhadap fitur menambahkan *rating* dan *review*.

7.1.11 Black Box Testing Fitur Simpan Resep

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 11, dan fitur pada gambar 4.24.

Tabel 7.11 Black box testing fitur simpan resep

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Simpan resep melalui halaman detail resep	Pengujian dengan MahasiswaKos membuka salah satu resep dan menklik icon simpan	Resep tersebut ditambahkan ke daftar resep tersimpan	Berhasil
2.	Simpan resep melalui icon simpan	Pengujian dengan MahasiswaKos menklik icon simpan pada salah satu resep pada daftar resep yang ditampilkan	Resep tersebut ditambahkan ke daftar resep tersimpan	Berhasil

Pada tabel 7.11 terdapat nama beserta deskripsi pengujian terhadap fitur simpan resep.

7.1.12 Black Box Testing Fitur Hapus Resep Tersimpan

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 12, dan fitur pada gambar 4.25.

Tabel 7.12 Black box testing fitur hapus resep tersimpan

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Hapus resep tersimpan melalui halaman detail resep	Pengujian dengan MahasiswaKos membuka salah satu resep dan	Resep tersebut dihapus dari daftar resep tersimpan	Berhasil

		menklik icon simpan pada resep yang telah tersimpan		
2.	Hapus resep tersimpan melalui icon simpan	Pengujian dengan MahasiswaKos menklik icon simpan pada salah satu resep pada resep yang telah tersimpan pada daftar resep yang ditampilkan	Resep tersebut dihapus dari daftar resep tersimpan	Berhasil

Pada tabel 7.12 terdapat nama beserta deskripsi pengujian terhadap fitur hapus resep tersimpan.

7.1.13 Black Box Testing Fitur Tambah Catatan Pribadi pada Resep

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 13, dan fitur pada gambar 4.26.

Tabel 7.13 Black box testing fitur tambah catatan pribadi pada resep

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Tambah catatan pribadi pada resep	Pengujian dengan MahasiswaKos membuka salah satu resep dan menambahkan catatan pribadi pada resep	Catatan pribadi tersebut ditambahkan dan tampil di resep tersebut	Berhasil

Pada tabel 7.13 terdapat nama beserta deskripsi pengujian terhadap fitur tambah catatan pribadi pada resep.

7.1.14 Black Box Testing Fitur Logout

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 14, dan fitur pada gambar 4.27.

Tabel 7.14 Black box testing fitur logout

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Logout	Pengujian dengan MahasiswaKos logout dari aplikasi melalui halaman pengaturan	Status pengguna yang menggunakan aplikasi berubah menjadi Tamu	Berhasil

Pada tabel 7.14 terdapat nama beserta deskripsi pengujian terhadap fitur logout.

7.1.15 Black Box Testing Fitur Minimum Android 5 Lollipop API 21

Berikut merupakan hasil *black box testing* yang mengacu pada *user scenario* yang terdapat pada tabel 4.7 nomor 15, dan fitur pada gambar 4.28.

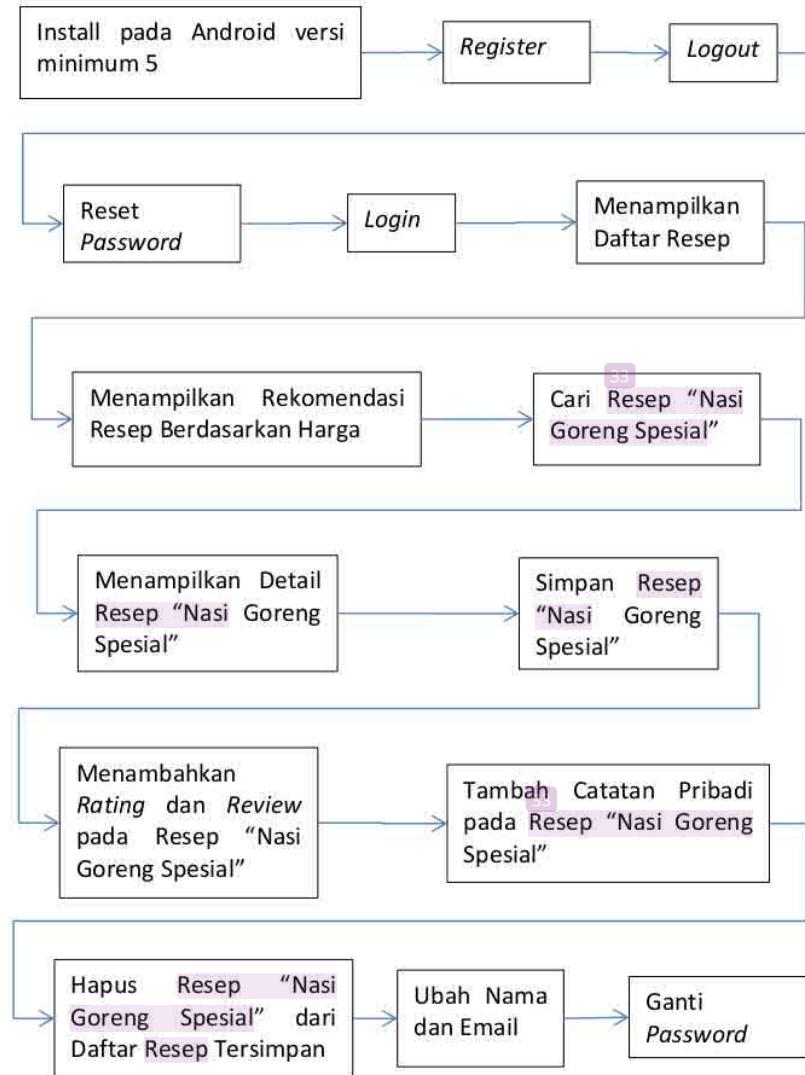
Tabel 7.15 Black box testing fitur minimum Android 5 Lollipop API 21

No	Nama Pengujian	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil
1.	Install aplikasi pada perangkat dengan minimum versi Android 5 Lollipop API 21	Pengujian dengan pengguna menginstall aplikasi pada perangkat Samsung Galaxy A52s dengan versi Android 11	Aplikasi telah terinstal dan dapat digunakan	Berhasil

Pada tabel 7.15 terdapat nama beserta deskripsi pengujian terhadap fitur minimum Android 5 Lollipop API 21.

7.2 Usability Testing

Usability testing merupakan sebuah pengujian untuk mengetahui apakah sebuah sistem telah memenuhi kebutuhan pengguna. *Usability testing* dilakukan menggunakan *scenario-based testing* secara *moderated usability testing* yang akan diuji kepada pengguna dengan ditemani moderator. Metode pengukuran *usability testing* yang digunakan pada penelitian ini yaitu efektivitas dan *System Usability Scale*. Pengujian dilakukan terhadap 10 partisipan yang dapat dilihat pada Lampiran D.1.



2

Gambar 7.1 Task scenario pada usability testing

Pada gambar 7.1 terdapat *task scenario* yang terdapat pada *usability testing* aplikasi ini. Skenario pada *scenario-based testing* menyerupai instruksi terkait bagaimana perangkat lunak digunakan.

Tabel 7.16 Deskripsi *task scenario* pada *usability testing*

No	Nama Skenario	Nama Fitur	Deskripsi Pengujian	Hasil yang Diharapkan
----	---------------	------------	---------------------	-----------------------

1.	Install Android pada versi minimum 5	Minimum Android 5 Lollipop API 21	Pengujian dengan pengguna menginstall aplikasi pada perangkat Samsung Galaxy A52s dengan versi Android 11	Aplikasi telah terinstal dan dapat digunakan
2.	<i>Register</i>	<i>Register</i>	Pengujian dengan Tamu mengisi nama, <i>email</i> yang belum pernah didaftarkan, dan <i>password</i> dan re-enter <i>password</i> dengan isi yang sama	Tamu terdaftar dan masuk ke sistem sebagai MahasiswaKos
3.	Logout	Logout	Pengujian dengan MahasiswaKos logout dari aplikasi melalui halaman pengaturan	Status pengguna yang menggunakan aplikasi berubah menjadi Tamu
4.	Reset <i>Password</i> dengan <i>email</i> yang didaftarkan	<i>Reset Password</i>	Pengujian dengan Tamu mengisi <i>email</i> untuk dikirimkan tautan reset <i>password</i> ke <i>email</i> tersebut	Tautan reset <i>password</i> terkirim ke <i>email</i> , dan kembali ke halaman <i>login</i>
5.	<i>Login</i>	<i>Login</i>	Pengujian dengan Tamu mengisi nama, <i>email</i> yang pernah didaftarkan, dan <i>password</i> yang sesuai	Tamu masuk ke sistem sebagai MahasiswaKos
6.	Menampilkan Daftar Resep	Menampilkan Daftar Resep	Pengujian dengan pengguna	Daftar resep ditampilkan

			membuka halaman daftar resep	
7.	Menampilkan Rekomendasi Resep Berdasarkan Harga	Menampilkan Rekomendasi Resep Berdasarkan Harga	Pengujian dengan pengguna membuka halaman daftar resep	Daftar resep masakan berdasarkan harga ditampilkan
8.	Cari Resep "Nasi Goreng Spesial"	Pencarian Resep	Pengujian dengan pengguna mengisi kolom pencarian dengan kata kunci yang benar dengan yang ada pada sistem dan menklik tombol cari	Daftar resep yang dicari ditampilkan
9.	Menampilkan Detail Resep "Nasi Goreng Spesial"	Menampilkan Detail Resep	Pengujian dengan pengguna membuka salah satu resep	Detail resep masakan ditampilkan
10.	Simpan Resep "Nasi Goreng Spesial"	Simpan Resep	Pengujian dengan MahasiswaKos membuka salah satu resep dan menklik icon simpan	Resep tersebut ditambahkan ke daftar resep tersimpan
11.	Menambahkan Rating dan Review pada Resep "Nasi Goreng Spesial"	Menambahkan Rating dan Review	Pengujian dengan MahasiswaKos membuka salah satu resep dan menambahkan rating dan review	Rating dan review tersebut ditambahkan dan ditampilkan di resep tersebut
12.	Tambah Catatan Pribadi pada Resep "Nasi Goreng Spesial"	Tambah Catatan Pribadi pada Resep	Pengujian dengan MahasiswaKos membuka salah satu resep dan	Catatan pribadi tersebut ditambahkan dan tampil di

			menambahkan catatan pribadi pada resep	resep tersebut
13.	Hapus Resep "Nasi Goreng Spesial" dari Daftar Resep Tersimpan	Hapus Resep Tersimpan	Pengujian dengan MahasiswaKos membuka salah satu resep dan menklik icon simpan pada resep yang telah tersimpan	Resep tersebut dihapus dari daftar resep tersimpan
14.	Ubah Nama dan Email	Ubah Profil	Pengujian dengan MahasiswaKos mengisi nama baru, dengan <i>email</i> baru yang belum terdaftar dalam sistem	Nama dan <i>Email</i> terbaru telah terubah dan kembali ke halaman pengaturan
15.	Ganti Password	Ganti Password	Pengujian dengan MahasiswaKos mengisi <i>password</i> lama dengan benar, dan <i>password</i> dan re-enter <i>password</i> dengan isi yang sama	<i>Password</i> berhasil terganti dan kembali ke halaman pengaturan

Pada tabel 7.16 terdapat deskripsi *task scenario* pada *usability testing*. Daftar *task scenario* yang dilakukan pada *usability testing* mengacu pada *user scenario* yang terdapat pada tabel 4.7, dan fitur pada subbab 4.3. Nama fitur juga mengacu kepada fitur yang terdapat pada subbab 4.3. Terdapat pula deskripsi dari daftar pengujian yang dilakukan beserta hasil yang diharapkan untuk nantinya diuji kepada partisipan.

7.2.1 Pengukuran Efektivitas *Usability Testing*

Berikut merupakan hasil *usability testing* yang mana masing-masing *task scenario* pengujian digambarkan pada gambar 7.1 serta dideskripsikan pada tabel 7.16, dan daftar partisipan dapat dilihat pada Lampiran D.1.

Tabel 7.17 Rekapitulasi hasil task scenario pada usability testing

No Partisipan	Nomor Task Scenario Pengujian														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v
2.	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v
3.	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v
4.	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v
5.	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v
6.	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v
7.	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v
8.	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v
9.	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v
10.	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v

Pada tabel 7.17 terdapat rekapitulasi hasil task scenario pada usability testing, dimana v adalah berhasil dan x adalah gagal.

Tabel 7.18 Rekapitulasi tingkat efektivitas usability testing

No Partisipan	Total task scenario berhasil	Tingkat efektivitas
1.	15	100%
2.	15	100%
3.	15	100%
4.	15	100%
5.	15	100%
6.	15	100%
7.	15	100%
8.	15	100%
9.	15	100%
10.	15	100%

Tingkat efektivitas keseluruhan (rata-rata tingkat efektivitas)	100%
---	------

Pada tabel 7.18 terdapat rekapitulasi tingkat efektivitas dari total 15 *task scenario* yang dihitung menggunakan rumus pada gambar 2.41.

7.2.2 Pengukuran *System Usability Scale*

Pengukuran *system usability scale* atau dapat disingkat *SUS* dilakukan menggunakan kuesioner yang diberikan kepada 10 responden yang hasilnya dapat dilihat pada Lampiran D.2. Berikut merupakan hasil pengukuran *SUS* yang mana masing-masing pertanyaan dapat dilihat pada tabel 2.5.

Tabel 7.19 Rekapitulasi nilai kuesioner *SUS*

No Responden	Pertanyaan										Nilai
	1	2	3	4	5	6	7	8	9	10	
1.	5	2	5	1	5	1	5	1	5	2	95
2.	3	1	5	1	4	2	5	1	5	1	90
3.	4	1	5	2	5	1	5	1	5	1	95
4.	4	2	4	1	4	1	4	1	5	1	87,5
5.	4	2	4	1	4	2	5	1	5	2	85
6.	4	2	4	1	5	1	5	1	5	4	85
7.	4	3	5	1	4	1	4	2	5	1	85
8.	4	2	5	4	5	2	3	2	4	3	70
9.	4	1	5	1	5	2	4	1	5	3	87,5
10.	4	2	4	3	4	2	4	2	4	1	75
Total											855
Nilai <i>SUS</i> (rata-rata/jumlah responden)											85,5

Pada tabel 7.19 terdapat rekapitulasi nilai kuesioner *SUS* dengan nilai rata-rata 85,5 yang dihitung menggunakan rumus pada subbab 2.18.4.

7.3 Kesimpulan Pengujian

Setelah melakukan pengujian atau *testing*, didapatkan beberapa hasil. Pada *black box testing* didapatkan hasil pengujian dengan tingkat keberhasilan 100%. Pada *usability testing* didapatkan hasil pengujian dengan tingkat efektivitas 100% yang didapat dari perhitungan jumlah task berhasil dibagi total task, lalu dikali

100%. Pada SUS didapatkan hasil pengujian dengan nilai 85,5, yang didapatkan dari perhitungan nilai pertanyaan instrumen dengan nomor ganjil dikurangi satu, perhitungan nilai pertanyaan instrumen dengan nomor genap memiliki perhitungan angka lima dikurangi nilai pertanyaan instrumen. Hasil penilaian per-pertanyaan akan menghasilkan skala 0 sampai 4 dengan 4 sebagai jawaban terbaik. Selanjutnya, penjumlahan jawaban dikali dengan 2,5, dan yang terakhir menghitung nilai rata-rata dari semua responden.

1 BAB 8 KESIMPULAN DAN SARAN

8.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, diperoleh beberapa kesimpulan yaitu:

1. Berdasarkan *persona* yang telah dibuat, maka dapat disimpulkan bahwa pengguna aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa adalah mahasiswa kos yang ingin memasak, dan mencari resep masakan menggunakan *smartphone*-nya.
2. Berdasarkan fitur-fitur yang telah dihasilkan maka dapat disimpulkan bahwa tujuan pengguna menggunakan aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa berbasis Android adalah melihat daftar resep, mencari resep, melihat daftar rekomendasi resep berdasarkan harga, melihat detail resep, menambahkan *rating* dan *review* terhadap sebuah resep, menyimpan resep, dan menambahkan catatan pribadi pada resep.
3. Berdasarkan hasil *black box testing*, didapatkan tingkat keberhasilan 100% yang berarti setiap fungsi pada aplikasi dapat dijalankan dengan baik dan sesuai dengan tugas yang diharapkan dari aplikasi. Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa dapat direalisasikan dalam bentuk aplikasi Android.
4. Berdasarkan hasil *usability testing* dan pengujian SUS yang dilakukan terhadap 10 responden, didapatkan tingkat efektivitas 100% dan nilai SUS 85,5. Tingkat efektivitas menggambarkan tingkat berhasil atau gagalnya partisipan dalam menjalankan *task scenario* pada aplikasi, yang berarti semua *task scenario* berhasil diselesaikan pengguna. Nilai 85,5 pada SUS dapat diinterpretasikan kepada penilaian *acceptability ranges*, *grade scale*, dan *adjective rating*. Untuk masing-masing nilai tersebut antara lain *acceptability ranges* adalah *acceptable* yang berarti perangkat lunak dapat diterima pengguna, *grade scale* adalah B yang berarti perangkat lunak memiliki *grade* atau tingkatan B, dan *adjective rating* adalah *excellent* yang berarti perangkat lunak memiliki *rating* istimewa. Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa pengguna mencapai tujuannya dengan menggunakan aplikasi yang dikembangkan.

1 8.2 Saran

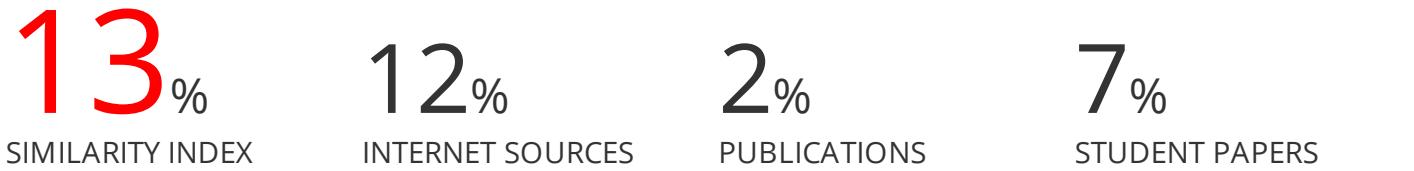
Berikut merupakan beberapa saran yang dapat dilakukan untuk mengembangkan penelitian ini lebih lanjut:

1. Menambahkan batasan jumlah *item* pada *recyclerview horizontal*, dan menambahkan tombol 'melihat lebih banyak' untuk melihat *item* pada *recyclerview horizontal* secara lengkap.
2. Menambahkan fitur video tutorial memasak.
3. Memperbaiki tombol *save* agar lebih menonjol.

4. Menambahkan halaman terpisah untuk menyimpan catatan-catatan pribadi untuk resep.

skripsi ziyad

ORIGINALITY REPORT



PRIMARY SOURCES

1	Submitted to Universitas Brawijaya Student Paper	5%
2	repository.ub.ac.id Internet Source	3%
3	repository.amikom.ac.id Internet Source	<1 %
4	repository.ittelkom-pwt.ac.id Internet Source	<1 %
5	www.scribd.com Internet Source	<1 %
6	repository.its.ac.id Internet Source	<1 %
7	repository.usd.ac.id Internet Source	<1 %
8	firebase.google.com Internet Source	<1 %
9	repository.dinamika.ac.id Internet Source	<1 %

10	www.coursehero.com	<1 %
Internet Source		
11	eprints.ums.ac.id	<1 %
Internet Source		
12	j-ptiik.ub.ac.id	<1 %
Internet Source		
13	repository.sar.ac.id	<1 %
Internet Source		
14	rekayasa.nusaputra.ac.id	<1 %
Internet Source		
15	Submitted to Sultan Agung Islamic University	<1 %
Student Paper		
16	jurnal.ipkia.ac.id	<1 %
Internet Source		
17	Submitted to UIN Sultan Syarif Kasim Riau	<1 %
Student Paper		
18	id.scribd.com	<1 %
Internet Source		
19	rutaprlia.wordpress.com	<1 %
Internet Source		
20	repository.uin-suska.ac.id	<1 %
Internet Source		
21	digilib.uinsgd.ac.id	<1 %
Internet Source		

22	vaskoedo.wordpress.com Internet Source	<1 %
23	Ramadhaniati Sari, Tursina Tursina, Anggi Srimurdianti Sukamto. "Aplikasi Pemilihan Resep Masakan Berdasarkan Ketersediaan Bahan Masakan dengan Metode Simple Matching Coefficient (SMC)", Jurnal Edukasi dan Penelitian Informatika (JEPIN), 2019 Publication	<1 %
24	Submitted to Universitas Jambi Student Paper	<1 %
25	digilibadmin.unismuh.ac.id Internet Source	<1 %
26	Submitted to Universitas Sebelas Maret Student Paper	<1 %
27	core.ac.uk Internet Source	<1 %
28	journal.trunojoyo.ac.id Internet Source	<1 %
29	www.mikroskil.ac.id Internet Source	<1 %
30	docplayer.info Internet Source	<1 %
31	ecampus.pelitabangsa.ac.id Internet Source	<1 %

32	media.neliti.com Internet Source	<1 %
33	resepasyiku.blogspot.com Internet Source	<1 %
34	library.binus.ac.id Internet Source	<1 %
35	repositori.uin-alauddin.ac.id Internet Source	<1 %
36	repository.unsri.ac.id Internet Source	<1 %
37	Submitted to Sriwijaya University Student Paper	<1 %
38	Submitted to Universitas Muria Kudus Student Paper	<1 %
39	eprints.usm.my Internet Source	<1 %
40	issuu.com Internet Source	<1 %
41	ojs.uajy.ac.id Internet Source	<1 %
42	Isa Bahroni, Riyadi Purwanto. "Aplikasi Pembelajaran (E-learning) Mengenal Huruf Hijaiyah bagi Anak-anak Berbasis Mobile untuk Mendukung Pembelajaran Secara	<1 %

Mandiri", Jurnal Edukasi dan Penelitian
Informatika (JEPIN), 2018

Publication

43	Submitted to Universitas Pendidikan Indonesia	<1 %
44	www.ista.ac.id Internet Source	<1 %
45	begawe.unram.ac.id Internet Source	<1 %
46	repository.uksw.edu Internet Source	<1 %
47	Submitted to Politeknik Negeri Jember Student Paper	<1 %
48	adoc.pub Internet Source	<1 %
49	ebin.pub Internet Source	<1 %
50	eprints.walisongo.ac.id Internet Source	<1 %
51	id.123dok.com Internet Source	<1 %
52	ms.desertanglican.org Internet Source	<1 %
	repository.uinsu.ac.id	

53	Internet Source	<1 %
54	sinta.unud.ac.id Internet Source	<1 %
55	tiniwarkam.blogspot.co.id Internet Source	<1 %
56	doaj.org Internet Source	<1 %
57	dspace.uii.ac.id Internet Source	<1 %
58	ejournal.nusamandiri.ac.id Internet Source	<1 %
59	repo.darmajaya.ac.id Internet Source	<1 %
60	repo.palcomtech.ac.id Internet Source	<1 %
61	text-id.123dok.com Internet Source	<1 %
62	www.justinmind.com Internet Source	<1 %
63	Fajar Abdul Aziz, Munawarah -, Primaadi Airlangga. "MOBILISASI PENYEBARAN INFORMASI KAMPUS BERBASIS FIREBASE CLOUD MESSAGING (FCM)", SAINTEKBU, 2019 Publication	<1 %

64

Rangga Gelar Guntara. "Aplikasi Pendekripsi
Penyakit Telinga Berbasis Android
menggunakan API Clarifai dan K-Nearest
Neighbor", Jurnal CoSciTech (Computer
Science and Information Technology), 2022

<1 %

Publication

65

digilib.uin-suka.ac.id

<1 %

Internet Source

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

On