PENGEMBANGAN APLIKASI PERANGKAT BERGERAK PANDUAN DAN RESEP MASAKAN UNTUK MAHASISWA BERBASIS ANDROID

SKRIPSI

Untuk memenuhi sebagian persyaratan   
memperoleh gelar Sarjana Komputer

Disusun oleh:

Muhammad Ziyad Al Maududi

NIM: 195150200111067



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2022

PENGESAHAN

Judul ini digunakan untuk dokumen final yang telah direvisi/disetujui setelah kelulusan ujian.

Untuk pendaftaran semhas dan ujian skripsi, judul halaman ini adalah **PERSETUJUAN.**

JUDUL SKRIPSI

SKRIPSI

Bagian ini hanya digunakan jika halaman ini adalah halaman **PENGESAHAN.** Jika ini adalah halaman PERSETUJUAN,maka bagian ini tidak diperlukan.

Diajukan untuk memenuhi sebagian persyaratan

memperoleh gelar Sarjana Komputer

Disusun Oleh :

Nama Mahasiswa

NIM: 123456789

Skripsi ini telah diuji dan dinyatakan lulus pada

2 Januari 2015

Telah diperiksa dan disetujui oleh:

|  |  |
| --- | --- |
| Dosen Pembimbing I  Nama Dosen Pembimbing 1  NIK: 123456789  /\*jika terdapat NIK saja\*/ | Dosen Pembimbing 2  Nama Dosen Pembimbing 2  NIK: -  /\*jika tidak terdapat NIP, NIK, atau keduanya\*/ |

Mengetahui

Ketua Jurusan **Nama Jurusan**

Contoh: Ketua Jurusan **Teknik** **Informatika**

Nama Ketua Jurusan

NIP: 123456789

/\*jika terdapat NIP\*/

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 1 Januari 2015



­

Nama Mahasiswa

NIM: 123456789

PRAKATA

Bagian ini memuat pernyataan resmi untuk menyampaikan rasa terima kasih penulis kepada berbagai pihak yang telah membantu penyelesaian skripsi ini. Nama-nama penerima ucapan terima kasih sebaiknya dituliskan lengkap, termasuk gelar akademik, dan pihak-pihak yang tidak terkait dihindari untuk dituliskan. Bahasa yang digunakan seharusnya mengikuti kaidah bahasa Indonesia yang baku. Prakata boleh diakhiri dengan paragraf yang menyatakan bahwa penulis menerima kritik dan saran untuk pengembangan penelitian selanjutnya. Terakhir, prakata ditutup dengan mencantumkan kota dan tanggal penulisan prakata, lalu diikuti dengan kata “Penulis”.

Malang, 1 Januari 2015

Penulis

email@domain.com

ABSTRAK

Nama Mahasiswa, Judul Skripsi

Pembimbing: Nama Pembimbing 1 dan Nama Pembimbing 2

Bagian ini diisi dengan abstrak dalam Bahasa Indonesia. Abstrak adalah uraian singkat (umumnya 200-300 kata) yang merupakan intisari dari sebuah skripsi. Abstrak membantu pembaca untuk mendapatkan gambaran secara cepat dan akurat tentang isi dari sebuah skripsi. Melalui abstrak, pembaca juga dapat menentukan apakah akan membaca skripsi lebih lanjut. Oleh karena itu, abstrak sebaiknya memberikan gambaran yang padat tetapi tetap jelas dan akurat tentang (1) apa dan mengapa penelitian dikerjakan: sedikit latar belakang, pertanyaan atau masalah penelitan, dan/atau tujuan penelitian; (2) bagaimana penelitian dikerjakan: rancangan penelitian dan metodologi/metode dasar yang digunakan dalam penelitian; (3) hasil penting yang diperoleh: temuan utama, karakteristik artefak, atau hasil evaluasi artefak yang dibangun; (4) hasil pembahasan dan kesimpulan: hasil dari analisis dan pembahasan temuan atau evaluasi artefak yang dibangun, yang dikaitkan dengan pertanyaan/tujuan penelitian.

Yang harus dihindari dalam sebuah abstrak diantaranya (1) penjelasan latar belakang yang terlalu panjang; (2) sitasi ke pustaka lainnya; (3) kalimat yang tidak lengkap; (3) singkatan, jargon, atau istilah yang membingungkan pembaca, kecuali telah dijelaskan dengan baik; (4) gambar atau tabel; (5) angka-angka yang terlalu banyak.

Di akhir abstrak ditampilkan beberapa kata kunci (normalnya 5-7) untuk membantu pembaca memposisikan isi skripsi dengan area studi dan masalah penelitian. Kata kunci, beserta judul, nama penulis, dan abstrak biasanya dimasukkan dalam basis data perpustakaan. Kata kunci juga dapat diindeks dalam basis data sehingga dapat digunakan untuk proses pencarian tulisan ilmiah yang relevan. Oleh karena itu pemilihan kata kunci yang sesuai dengan area penelitian dan masalah penelitian cukup penting. Pemilihan kata kunci juga bisa didapatkan dari referensi yang dirujuk.

Kata kunci: abstrak, skripsi, intisari, kata kunci, artefak

ABSTRACT

Student Name, Skripsi Title

Supervisors: First Supervisor’s Name and Second Supervisor’s Name

The absract of your skripsi in English is written here.

DAFTAR ISI

[PENGESAHAN ii](#_Toc115268981)

[PERNYATAAN ORISINALITAS iii](#_Toc115268982)

[PRAKATA iv](#_Toc115268983)

[ABSTRAK v](#_Toc115268984)

[ABSTRACT vi](#_Toc115268985)

[DAFTAR ISI vii](#_Toc115268986)

[DAFTAR TABEL x](#_Toc115268987)

[DAFTAR GAMBAR xi](#_Toc115268988)

[DAFTAR LAMPIRAN xii](#_Toc115268989)

[BAB 1 PENDAHULUAN 1](#_Toc115268990)

[1.1 Latar Belakang 1](#_Toc115268991)

[1.2 Rumusan Masalah 3](#_Toc115268992)

[1.3 Tujuan 3](#_Toc115268993)

[1.4 Manfaat 3](#_Toc115268994)

[1.5 Batasan Masalah 4](#_Toc115268995)

[1.6 Sistematika Pembahasan 4](#_Toc115268996)

[BAB 2 LANDASAN KEPUSTAKAAN 5](#_Toc115268997)

[2.1 Kajian Pustaka 5](#_Toc115268998)

[2.2 Memasak 5](#_Toc115268999)

[2.2.1 Resep Masakan 5](#_Toc115269000)

[2.3](#_Toc115269001) *[Prototyping](#_Toc115269001)* [6](#_Toc115269001)

[2.4](#_Toc115269002) *[Personas](#_Toc115269002)* [7](#_Toc115269002)

[2.5](#_Toc115269003) *[Scenarios](#_Toc115269003)* [9](#_Toc115269003)

[2.6](#_Toc115269004) *[User Stories](#_Toc115269004)* [10](#_Toc115269004)

[2.7](#_Toc115269005) *[Features](#_Toc115269005)* [11](#_Toc115269005)

[2.8 Aplikasi Perangkat Bergerak 16](#_Toc115269006)

[2.9 Android 16](#_Toc115269007)

[2.10 Android Studio 17](#_Toc115269008)

[2.11 Firebase 17](#_Toc115269009)

[2.12 Arsitektur Pengembangan Aplikasi Android 19](#_Toc115269010)

[2.13 Pengujian Perangkat Lunak 21](#_Toc115269011)

*[2.13.1 Scenario-Based Testing](#_Toc115269012)* [21](#_Toc115269012)

[2.13.2](#_Toc115269013) *[Usability](#_Toc115269013)**[Testing](#_Toc115269013)* [23](#_Toc115269013)

[BAB 3 METODOLOGI 26](#_Toc115269014)

[3.1 Tipe Penelitian 26](#_Toc115269015)

[3.2 Metode Penelitian 26](#_Toc115269016)

[3.3 Subjek Penelitian 28](#_Toc115269017)

[3.4 Lokasi Penelitian 28](#_Toc115269018)

[3.5 Teknik Pengumpulan Data 28](#_Toc115269019)

[3.6 Peralatan Pendukung yang Digunakan 28](#_Toc115269020)

[3.7 Metode Pengembangan 28](#_Toc115269021)

[3.7.1 Studi Literatur, Survey, Wawancara 29](#_Toc115269022)

*[3.7.2](#_Toc115269023)* [Analisis Kebutuhan 29](#_Toc115269023)

[3.7.3 Pengembangan](#_Toc115269024) *[Prototype](#_Toc115269024)* [30](#_Toc115269024)

*[3.7.4](#_Toc115269025)* [Evaluasi](#_Toc115269025) *[Prototype](#_Toc115269025)* [30](#_Toc115269025)

[3.7.5 Perancangan 31](#_Toc115269026)

[3.7.6 Implementasi 31](#_Toc115269027)

*[3.7.7](#_Toc115269028)* [Pengujian 31](#_Toc115269028)

[3.7.8 Kesimpulan dan Saran 32](#_Toc115269029)

[BAB 4 HASIL 33](#_Toc115269030)

[4.1 Subbab Dua Satu 33](#_Toc115269031)

[4.2 Subbab Dua Dua 33](#_Toc115269032)

[4.2.1 Subbab Empat Dua Satu 33](#_Toc115269033)

[4.2.2 Subbab Empat Dua Dua 34](#_Toc115269034)

[4.3 Subbab Empat Tiga 34](#_Toc115269035)

[4.3.1 Contoh Struktur Penelitian Implementatif Pengembangan 34](#_Toc115269036)

[4.3.2 Contoh Struktur Penelitian Nonimplementatif 35](#_Toc115269037)

[BAB 5 PEMBAHASAN 36](#_Toc115269038)

[5.1 Subbab Lima Satu 36](#_Toc115269039)

[5.1.1 Subbab Lima Satu Satu 36](#_Toc115269040)

[5.1.2 Subbab Lima Satu Dua 36](#_Toc115269041)

[5.2 Subbab Lima Dua 37](#_Toc115269042)

[5.2.1 Subbab Lima Dua Satu 37](#_Toc115269043)

[5.2.2 Subbab Lima Dua Dua 37](#_Toc115269044)

[5.3 Subbab Lima Tiga 37](#_Toc115269045)

[5.3.1 Contoh Struktur Penelitian Implementatif Pembangunan 37](#_Toc115269046)

[5.3.2 Contoh Struktur Penelitian Nonimplementatif Eksperimental 39](#_Toc115269047)

[BAB 6 Penutup 40](#_Toc115269048)

[6.1 Kesimpulan 40](#_Toc115269049)

[6.2 Saran 40](#_Toc115269050)

[DAFTAR REFERENSI 41](#_Toc115269051)

[LAMPIRAN A KUESIONER TENTANG RESEP MASAKAN 46](#_Toc115269052)

[A.1 Kuesioner Masyarakat Umum Tentang Resep Masakan 46](#_Toc115269053)

[A.2 Kuesioner Mahasiswa Filkom Universitas Brawijaya Tentang Resep Masakan 49](#_Toc115269054)

DAFTAR TABEL

[Tabel 2.1 Dependensi Firebase Authentication 18](#_Toc115268554)

[Tabel 2.2 Dependensi FirebaseUI 18](#_Toc115268555)

[Tabel 2.3 Dependensi Firebase Cloud Firestore 18](#_Toc115268556)

[Tabel 2.4 Dependensi Firebase Cloud Storage 19](#_Toc115268557)

[Tabel 2.5 Instrumen pengujian](#_Toc115268558) *[System Usability Scale](#_Toc115268558)* [23](#_Toc115268558)

[Tabel 3.1 Daftar Pertanyaan Wawancara untuk Analisis Kebutuhan 29](#_Toc115268559)

[Tabel 3.2 Daftar Pertanyaan Wawancara untuk Evaluasi](#_Toc115268560) *[Prototype](#_Toc115268560)* [30](#_Toc115268560)

[Tabel 3.3 Daftar Pertanyaan Kuesioner untuk Pengujian 32](#_Toc115268561)

DAFTAR GAMBAR

[Gambar 2.1 Tahapan dari](#_Toc115268562) *[Personas](#_Toc115268562)* [hingga menjadi](#_Toc115268562) *[Features](#_Toc115268562)* [7](#_Toc115268562)

[Gambar 2.2 Deskripsi sebuah](#_Toc115268563) *[Personas](#_Toc115268563)* [8](#_Toc115268563)

[Gambar 2.3 Contoh](#_Toc115268564) *[Personas](#_Toc115268564)* [dalam bentuk deskripsi naratif 9](#_Toc115268564)

[Gambar 2.4 Deskripsi sebuah](#_Toc115268565) *[Scenarios](#_Toc115268565)* [9](#_Toc115268565)

[Gambar 2.5 Contoh](#_Toc115268566) *[Scenarios](#_Toc115268566)* [10](#_Toc115268566)

[Gambar 2.6 Format standar kalimat](#_Toc115268567) *[User Stories](#_Toc115268567)* [11](#_Toc115268567)

[Gambar 2.7 Contoh kalimat](#_Toc115268568) *[User Stories](#_Toc115268568)* [dengan format standar 11](#_Toc115268568)

[Gambar 2.8 Format varian standar kalimat](#_Toc115268569) *[User Stories](#_Toc115268569)* [dengan justifikasi 11](#_Toc115268569)

[Gambar 2.9 Contoh kalimat](#_Toc115268570) *[User Stories](#_Toc115268570)* [dengan format varian standar 11](#_Toc115268570)

[Gambar 2.10 Contoh](#_Toc115268571) *[User Stories](#_Toc115268571)* [dari sebuah](#_Toc115268571) *[Scenarios](#_Toc115268571)* [11](#_Toc115268571)

[Gambar 2.11 Deskripsi sebuah](#_Toc115268572) *[Features](#_Toc115268572)* [12](#_Toc115268572)

[Gambar 2.12 Format kalimat deskripsi sebuah](#_Toc115268573) *[Features](#_Toc115268573)* [12](#_Toc115268573)

[Gambar 2.13 Desain](#_Toc115268574) *[Features](#_Toc115268574)* [13](#_Toc115268574)

[Gambar 2.14 Faktor dalam set desain](#_Toc115268575) *[Features](#_Toc115268575)* [14](#_Toc115268575)

[Gambar 2.15 Pertanyaan terhadap](#_Toc115268576) *[Features](#_Toc115268576)* [untuk menghindari](#_Toc115268576) *[feature creep](#_Toc115268576)* [14](#_Toc115268576)

[Gambar 2.16 Melakukan](#_Toc115268577) *[highlight](#_Toc115268577)* [frasa terhadap](#_Toc115268577) *[Scenarios](#_Toc115268577)* [15](#_Toc115268577)

[Gambar 2.17 Contoh penentuan](#_Toc115268578) *[Features](#_Toc115268578)* [berdasarkan](#_Toc115268578) *[Scenarios](#_Toc115268578)* [15](#_Toc115268578)

[Gambar 2.18 Contoh](#_Toc115268579) *[Features](#_Toc115268579)* [menggunakan template model](#_Toc115268579) *[input/action/output](#_Toc115268579)* [16](#_Toc115268579)

[Gambar 2.19 Komponen aplikasi Android 17](#_Toc115268580)

[Gambar 2.20 Rancangan arsitektur](#_Toc115268581) *[MVVM](#_Toc115268581)* [21](#_Toc115268581)

[Gambar 2.21 Contoh skenario pada](#_Toc115268582) *[scenario-based testing](#_Toc115268582)* [22](#_Toc115268582)

[Gambar 2.22 Contoh sebuah](#_Toc115268583) *[scenario-based testing](#_Toc115268583)* [22](#_Toc115268583)

[Gambar 2.23 Skala penilaian](#_Toc115268584) *[System Usability Scale](#_Toc115268584)* [24](#_Toc115268584)

[Gambar 2.24 Langkah-langkah](#_Toc115268585) *[System Usability Scale](#_Toc115268585)* [24](#_Toc115268585)

[Gambar 2.25 Penilaian](#_Toc115268586) *[System Usability Scale](#_Toc115268586)* [25](#_Toc115268586)

[Gambar 3.1 Metode penelitian 27](#_Toc115268587)

DAFTAR LAMPIRAN

[LAMPIRAN A KUESIONER TENTANG RESEP MASAKAN 46](#_Toc115269052)

[A.1 Kuesioner Masyarakat Umum Tentang Resep Masakan 46](#_Toc115269053)

[A.2 Kuesioner Mahasiswa Filkom Universitas Brawijaya Tentang Resep Masakan 49](#_Toc115269054)

# PENDAHULUAN

## 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 (Telset, 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 persen responden pernah, terkadang, atau suka memasak, didapatkan hasil bahwa 95,3 persen responden belajar memasak dari resep maupun tutorial, baik dari internet, turun-temurun, diajarkan orang lain, maupun dari buku resep. 79,2 persen 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 persen 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 persen responden pernah, terkadang, atau suka memasak, didapatkan hasil bahwa 80,5 persen 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 persen 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 persen responden merasa akan terbantu dalam mengatasi permasalahan terkait memasak dengan adanya aplikasi yang memberikan resep masakan.

Berdasarkan penelitian yang dilakukan Ichwan, Sudjalwo, dan Supardi (2013), mereka membuat penelitian yang berjudul “Aplikasi Resep Masakan Khas Jawa Berbasis Android”. Pengembangan dalam penelitian tersebut menggunakan metode *SDLC* *Waterfall*, yang mana menghasilkan sebuah aplikasi yang membantu dalam memasak, dan cukup efisien dalam penggunaannya, dengan tampilan aplikasi yang cukup baik.

Penelitian lain pernah dilakukan oleh 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.

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 pengguna, dalam aplikasi ini yaitu mahasiswa, dapat memasak masakan sederhana dengan bahan yang mudah didapat, resep dan tingkat kematangan yang sesuai, dan hasil yang diinginkan. Nantinya aplikasi perangkat bergerak ini akan diuji, apakah telah memenuhi kebutuhan pengguna atau tidak.

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, Kharisma, & Arwani, 2022).

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 (Kementrian 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).**(kenapa android di lingkungan penelitian)**

Pengembangan Aplikasi Perangkat Bergerak Panduan dan Resep Masakan untuk Mahasiswa berbasis Android diharapkan mampu menyelesaikan permasalahan-permasalahan yang dialami mahasiswa yang suka memasak.

## 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?

## 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.

## Manfaat

Manfaat dari penelitian pengembangan aplikasi perangkat bergerak panduan dan resep masakan untuk mahasiswa berbasis Android adalah sebagai berikut: **(manfaat aplikasi untuk siapa, untuk mhs)**

1. Untuk Mahasiswa:

a. Memberikan resep dan panduan dalam memasak, dalam hal ini masakan sederhana yang sesuai dengan mahasiswa

## 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.**(dipertanggungjawabkan kenapa v23 dan ga lbh rendah) <https://developer.android.com/about/versions/lollipop/android-5.0>**
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.

## 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.

1. 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.

1. BAB 3 – METODOLOGI PENELITIAN

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

1. BAB 4 - ANALISIS KEBUTUHAN

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

1. BAB 5 – PERANCANGAN

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

1. BAB 6 – IMPLEMENTASI

Babi ini berisi tentang implementasi dari perancangan perangkat lunak.

1. BAB 7 – PENGUJIAN

Bab ini berisi tentang pengujian dari perangkat lunak.

1. BAB 8 – PENUTUP

Bab ini berisi tentang kesimpulan dan saran.

# LANDASAN KEPUSTAKAAN

## 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, Kharisma, & Arwani, 2022).**(buat tabel)**

|  |  |  |  |
| --- | --- | --- | --- |
| Penulis | Judul | Metode | Hasil |
| [Lina](https://etd.unsyiah.ac.id/index.php?author=%22Tomy+Inri+Akbar+Lingga%22&search=Search) Karlina  Jelita Asian  Muhamad Mahmud | Rancang Bangun Aplikasi Resep Masakan Menggunakan Metode Xp Berbasis Android (2019) | Metode *SDLC Extreme Programming* | Aplikasi membantu masyarakat mengetahui resep dan tatacara pembuatan masakan dengan mudah |
| Ilham Dwi Muchlison  Agi Putra Kharisma  Issa Arwani | Pengembangan Aplikasi Perangkat Bergerak Sistem Informasi Event di Bidang Teknologi Informasi Berbasis Android (2021) | Metode *SDLC 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 menyebarkan informasi terkait event yang diadakan dengan menjangkau peserta yang sesuai dengan topik dibidangnya.. |
| Pristi Sukmasetya  Agus Setiawan  Endah Ratna Arumi | Penggunaan *Usability Testing* Sebagai Alat Evaluasi Website KRS *Online* pada Perguruan Tinggi | Metode *Usability Testing* | Penelitian ini mengevaluasi *website* KRS o*nline* yang terdapat pada perguruan tinggi dengan *in depth interview* dan *usability testing* untuk mengetahui tingkat kebergunaan dari sistem tersebut. |

Terdapat beberapa penelitian terdahulu yang berhubungan dengan aplikasi resep masakan. Berdasarkan penelitian yang 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*, dengan bahasa pemrograman yang digunakan yaitu Delphi. Penelitian tersebut 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 *SDLC* *Prototyping*.

## 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).

### 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).

## *SDLC*

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).

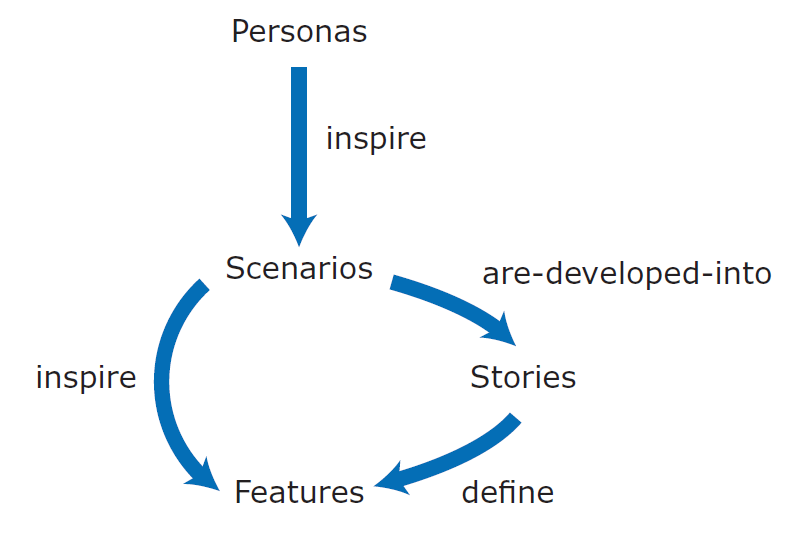
*Software Development Life Cycle* atau *SDLC* adalah sebuah metode pengembangan perangkat lunak yang berisi analisis kebutuhan perangkat lunak, desain, pembuatan kode program, pengujian, dan pemeliharaan. Dalam pengembangan perangkat lunak, dibutuhkan *SDLC* untuk mengurangi terjadina krisis perangkat 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).

## *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. Sebelum membuat *Prototype* pelanggan, dibutuhkan beberapa studi terhadap pengguna dan ide yang jelas mengenai pengguna yang potensial, dan skenario yang digunakan. Untuk itu, dibutuhkan *Personas* dan *Scenarios*, yang mana nantinya akan menghasilkan *User Stories* untuk dipakai dalam penentuan *Features*. 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).



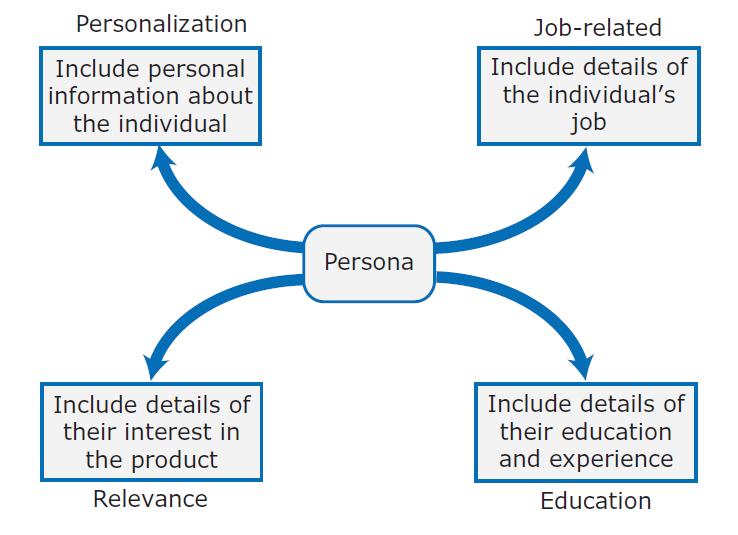
Gambar 2.1 Tahapan dari *Personas* hingga menjadi *Features*

Sumber: Sommerville (2021)

## *Personas*

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 *Personas* untuk merepresentasikan *shared vision* tersebut (Sommerville, 2021).

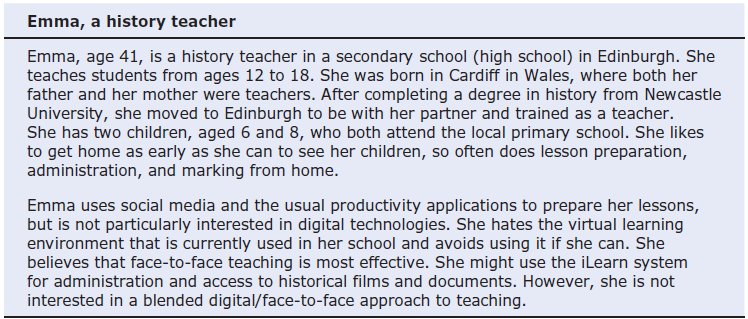
*Personas* adalah sebuah ‘pengguna khayalan’, potret karakter dari tipe pengguna yang mungkin akan menggunakan produk tersebut. Sebagai contoh, pada sebuah aplikasi pemesanan dokter gigi, akan terdapat 3 macam *Personas* yaitu dokter gigi, resepsionis, dan pasien. *Personas* untuk tipe pengguna yang berbeda akan membantu membayangkan apa saja yang mungkin pengguna inginkan dari perangkat lunak, bagaimana mereka menggunakannya, dan kesulitan yang mungkin akan dialami dalam memahami dan menggunakan fitur. Beberapa rekomendasi aspek dari sebuah deskripsi persona antara lain personalisasi, relevansi, edukasi, dan pekerjaan. Aspek personalisasi merupakan sebuah informasi dan keadaan personal dari sebuah *Personas.* Aspek pekerjaan merupakan sebuah informasi detail terkait pekerjaan dari sebuah *Personas.* Aspek Edukasi merupakan sebuah informasi terkait edukasi, kemampuan teknis, dan pengalaman dari sebuah *Personas.* 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).



Gambar 2.2 Deskripsi sebuah *Personas*

Sumber: Sommerville (2021)

Dalam membuat sebuah *Personas*, 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 *Personas*. *Personas* yang dibuat harus diperiksa ulang terhadap data pengguna untuk memastikan bahwa *Personas* tersebut mencerminkan tipikal pengguna produk (Sommerville, 2021). Sebelum nantinya membuat *User Stories*, diharuskan untuk memahami target pengguna dengan menggunakan *Personas*. Hal itu karena pada *User Stories* menceritakan kisah pengguna dalam menggunakan produk, sehingga dapat diketahui siapa pengguna, masalah yang ingin dipecahkan, dan daftar keinginan dari deskripsi fungsi produk yang relevan. *Personas* menawarkan cara yang bagus dalam menggambarkan pengguna dengan kebutuhannya (Romanpichler, 2014).

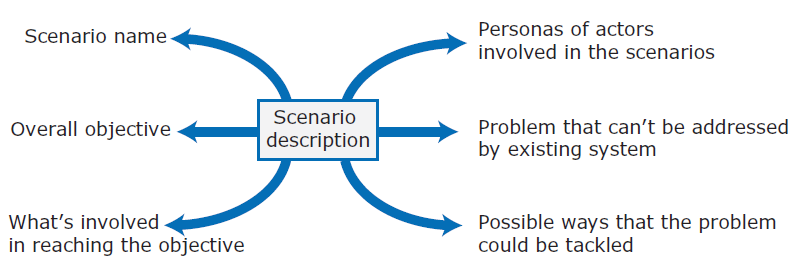


Gambar 2.3 Contoh *Personas* dalam bentuk deskripsi naratif

Sumber: Sommerville (2021)

## *Scenarios*

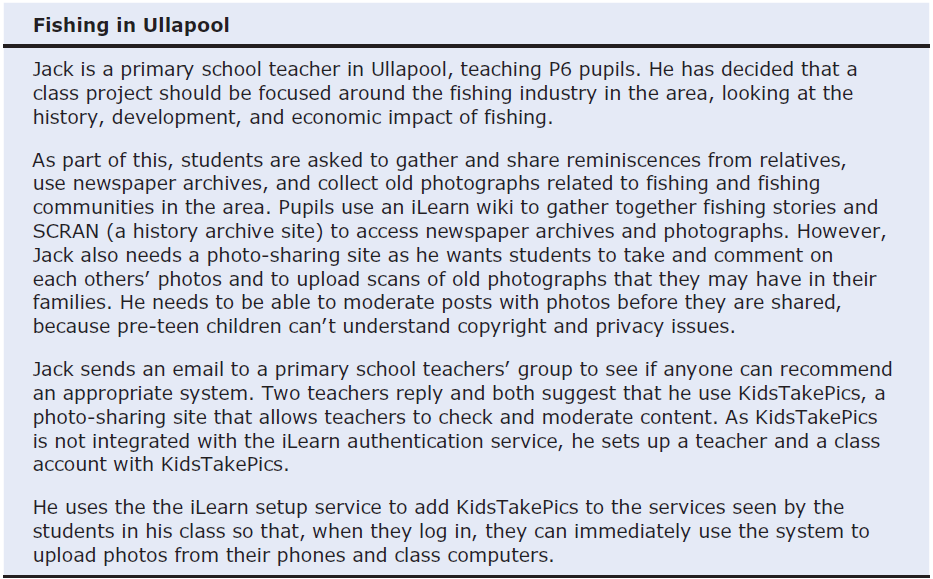
*Scenarios* adalah sebuah narasi mengenai situasi yang mana pengguna menggunakan *Features* atau fitur dari produk untuk melakukan hal yang mereka inginkan. Secara singkat, *Scenarios* haruslah memaparkan masalah pengguna, dan menunjukkan cara yang dibayangkan dalam penyelesaian masalah tersebut. Menyertakan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



Gambar 2.4 Deskripsi sebuah *Scenarios*

Sumber: Sommerville (2021)

*Scenarios* digunakan dalam tahap desain persyaratan dan fitur sistem, pengujian, dan desain antarmuka pengguna. *Scenarios* 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 *Scenarios* yang biasa dipakai, yaitu *Scenarios* naratif, dan *Scenarios* struktural. *Scenarios* naratif lebih direkomendasikan karena lebih mudah untuk dimengerti oleh pengguna sistem. Sebuah *Scenarios* 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).



Gambar 2.5 Contoh *Scenarios*

Sumber: Sommerville (2021)

## *User Stories*

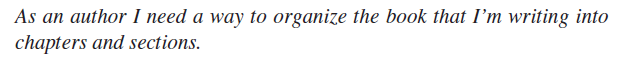
*User Stories* adalah sebuah deskripsi mengenai situasi dimana pengguna mencoba melakukan sesuatu dengan sistem perangkat lunak. *Scenarios* merupakan *user* *stories* tingkat tinggi yang dipakai oleh sistem. *User Stories* haruslah menjelaskan urutan interaksi dengan sistem, namun tidak termasuk detail dari interaksi tersebut. Selain itu, *User Stories* 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 Stories* adalah untuk perencanaan, dimana *User Stories* harus berfokus pada kejelasan fitur sistem yang ditentukan, atau aspek dari fitur yang dapat diimplementasikan. Saat menentukan *User Stories* dari sebuah *Scenarios*, lebih banyak informasi akan disediakan untuk pengembang untuk membantu mereka mendesain fitur dari produk. *Scenarios* dan *User Stories* sangat berguna dalam memilih dan mendesain fitur pada sistem, namun hanya sebagai ‘alat untuk berfikir’, dan bukan sebagai spesifikasi sistem. *Scenarios* dan *User Stories* hanya dipakai untuk menstimulasikan cara berfikir, yang mana tidak harus selesai ataupun konsisten, dan tidak ada aturan tetap terkait berapa banyak *Scenarios* dan *User Stories* yang dibutuhkan.



Gambar 2.6 Format standar kalimat *User Stories*

Sumber: Sommerville (2021)



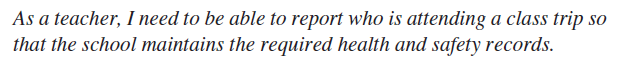
Gambar 2.7 Contoh kalimat *User Stories* dengan format standar

Sumber: Sommerville (2021)



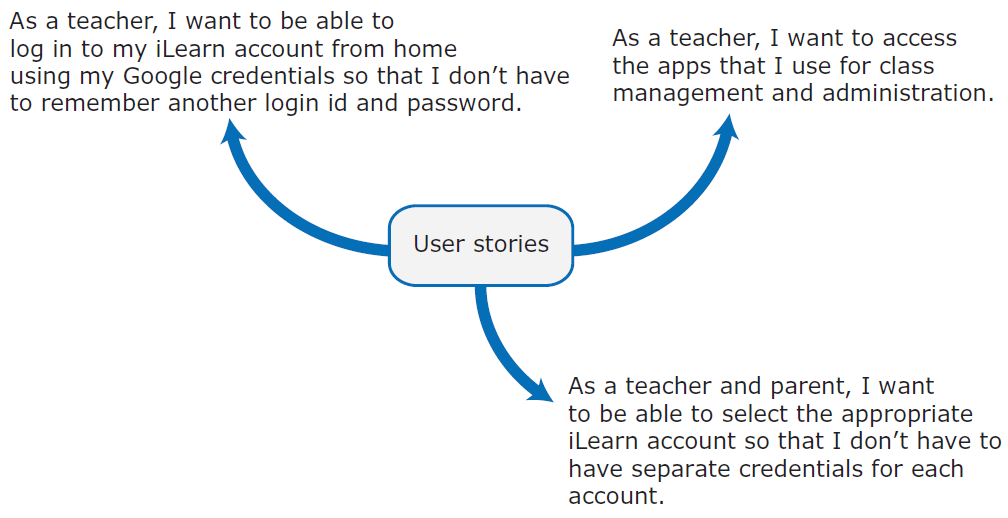
Gambar 2.8 Format varian standar kalimat *User Stories* dengan justifikasi

Sumber: Sommerville (2021)



Gambar 2.9 Contoh kalimat *User Stories* dengan format varian standar

Sumber: Sommerville (2021)

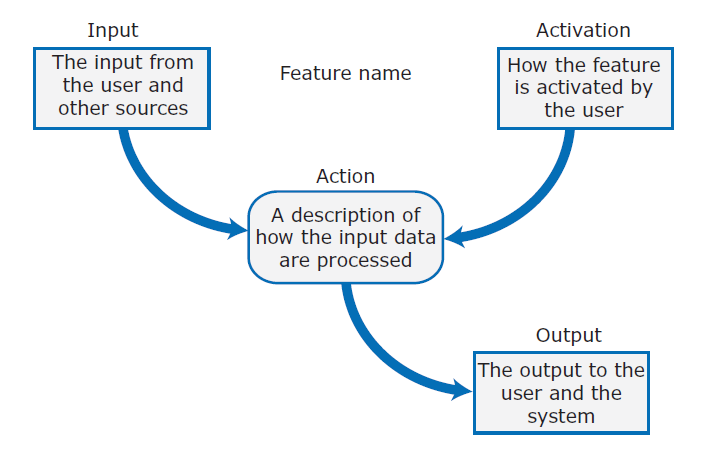


Gambar 2.10 Contoh *User Stories* dari sebuah *Scenarios*

Sumber: Sommerville (2021)

## *Features*

*Features* atau fitur adalah sebuah cara yang memungkinkan pengguna mengakses dan menggunakan fungsionalitas produk, sehingga daftar fitur mendefinisikan fungsionalitas sistem secara keseluruhan. Mengidentifikasikan 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 2.11 Deskripsi sebuah *Features*

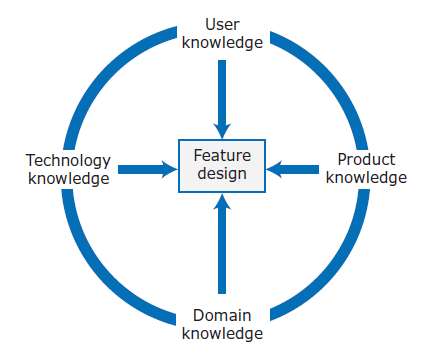
Sumber: Sommerville (2021)



Gambar 2.12 Format kalimat deskripsi sebuah *Features*

Sumber: Sommerville (2021)

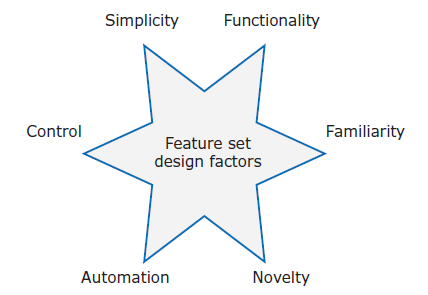
Tidak ada metode pasti untuk pemilihan dan pendesainan sebuah fitur. Namun, terdapat 4 sumber pengetahuan penting yang dapat membantu dalam memilih dan mendesain sebuah fitur, yaitu pengetahuan pengguna, pengetahuan produk, pengetahuan domain, dan pengetahuan teknologi. Pada pengetahuan pengguna, dapat digunakan *Scenarios* dan *User Stories* untuk memberitahu tim mengenai apa yang diinginkan pengguna dan bagaimana mereka menggunakan fitur dari perangkat lunak. Pada pengetahuan produk, mungkin telah dimiliki pengalaman dengan produk yang telah ada, atau memutuskan untuk meneliti produk tersebut sebagai bagian dari pengembangan produk yang dikembangkan, bahkan terkadang fitur dasarnya perlu untuk ditiru. Pada pengetahuan domain, hal ini merupakan terkait dengan domain atau area kerja yang ingin didukung oleh produk yang dikembangkan, sehingga dapat menemukan cara baru yang inovatif dalam membantu pengguna melakukan hal yang diinginkan. Pada pengetahuan teknologi, sebuah produk baru sering muncul untuk mengambil keuntungan dari pengembangan teknologi sejak kompetitor hadir, dimana memanfaatkan teknologi terbaru dalam mendesain fitur yang menggunakannya. Namun, tidak semuanya sebegitu penting untuk semua jenis produk. Misalnya, pengetahuan domain sangat penting untuk produk bisnis, namun kurang penting untuk produk konsumen generik (Sommerville, 2021).



Gambar 2.13 Desain *Features*

Sumber: Sommerville (2021)

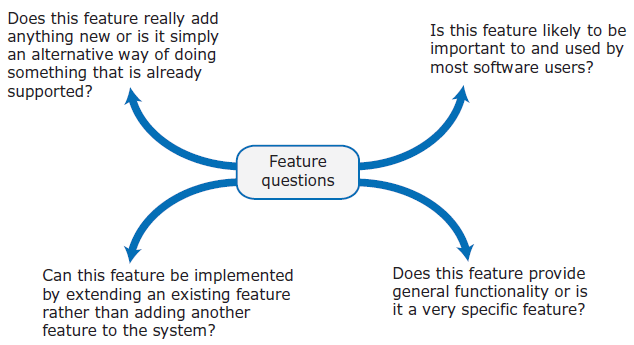
Ada 6 faktor yang perlu dipertimbangkan saat mendesain sebuah set fitur dari produk, dan menentukan bagaimana fitur-fitur tersebut bekerja. Namun, mustahil untuk menerapkan semua faktor tersebut secara optimal, sehingga diperlukan sejumlah pertukaran. Pertukaran-pertukaran dalam faktor tersebut antara lain *simplicity and functionality*, *familiarity and novelty*, dan *automation and control*. *Simplicity and functionality* berarti menemukan keseimbangan antara menyediakan kesederhanaan, kemudahan penggunaan sistem, dan termasuk fungsionalitas yang cukup untuk memikat pengguna dengan keinginan yang beragam. *Familiarity and novelty* berarti menyediakan fitur baru yang meyakinkan pengguna bahwa produk yang dibuat dapat melakukan hal yang lebih bisa dilakukan dari kompetitor. *Automation and control* berarti berfikir dengan hati-hati dalam menentukan apa yang dapat diotomatisasi, bagaimana mengotomatisasinya, dan bagaimana pengguna dapat mengatur otomatisasi sehingga sistem dapat disesuaikan dengan preferensi pengguna (Sommerville, 2021).



Gambar 2.14 Faktor dalam set desain *Features*

Sumber: Sommerville (2021)

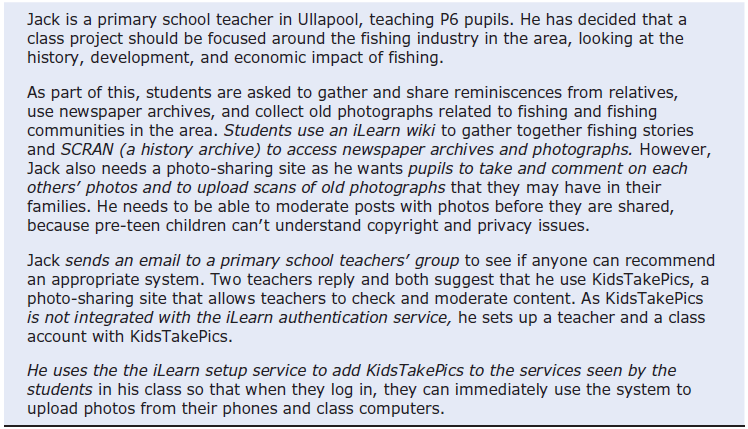
Terdapat satu masalah yang harus dihindari pengembang yaitu ‘*feature creep*’. *Feature creep* adalah beberapa fitur dari produk yang merayap naik sebagai potensi penggunaan baru dari produk. Hal ini dapat memperumit kompleksitas produk, yang mana dapat mendatangkan *bugs* dan kerentanan keamanan ke dalam perangkat lunak. Selain itu, *feature creep* dapat membuat antarmuka menjadi lebih kompleks, menu bertingkat yang lebih tinggi, yang mana dapat membingungkan pengguna. Untuk menghindarinya, *product manager* dan tim pengembang harus meninjau semua proposal fitur, dan membandingkannya dengan fitur yang telah diterima untuk diimplementasikan (Sommerville, 2021).



Gambar 2.15 Pertanyaan terhadap *Features* untuk menghindari *feature creep*

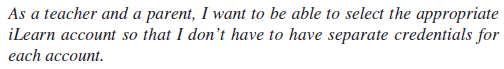
Sumber: Sommerville (2021)

*Features* bisa didapatkan dari melakukan *highlight* terhadap deskripsi naratif pada *Scenarios* untuk menemukan fitur sistem. Caranya yaitu membaca *Scenarios*, 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 mengeneralisir set dari fitur produk. Jika *User Stories* dikembangkan untuk menyempurnakan *Scenarios*, maka fitur produk atau karakteristik fitur dapat lebih mudah didapatkan dari *Scenarios* tersebut (Sommerville, 2021).



Gambar 2.16 Melakukan *highlight* frasa terhadap *Scenarios*

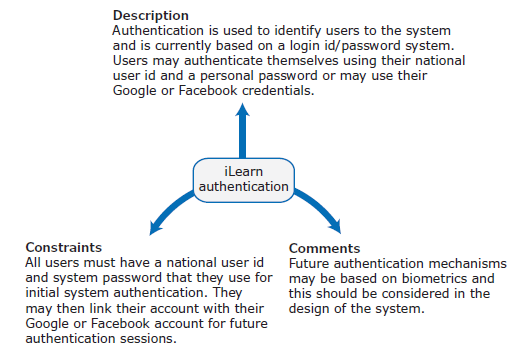
Sumber: Sommerville (2021)



Gambar 2.17 Contoh penentuan *Features* berdasarkan *Scenarios*

Sumber: Sommerville (2021)

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 2.18 Contoh *Features* menggunakan template model *input/action/output*

Sumber: Sommerville (2021)

## Aplikasi Perangkat Bergerak

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).

## 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).

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).



Gambar 2.19 Komponen aplikasi Android

Sumber: Abdelbaki, Charkaoui, & Lahmar (2016)

## Android Studio

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

## Firebase

Firebase adalah sebuah *BaaS* atau *Backend 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 Backend (Nugraha, 2019). Firebase sendiri menyediakan beberapa layanan, antara lain Firebase Authentication, Firebase Cloud Firestore, dan Firebase Cloud Storage (Firebase, 2022a).

#### 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 kata sandi, 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 2.1 Dependensi Firebase Authentication

|  |
| --- |
| dependencies {  // Impor BoM untuk *platform* Firebase  implementation *platform*('com.google.firebase: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.firebase:firebase-auth-ktx'  } |

Sumber: Firebase (2022c)

#### FirebaseUI

FirebaseUI adalah sebuah layanan autentikasi yang dapat dikustomisasi, *open source, drop-in* yang dapat menangani alur antarmuka pengguna untuk autentikasi masuknya pengguna. Praktik terbaik untuk autentikasi pada aplikasi perangkat bergerak diterapkan pada komponen dari FirebaseUI, guna memaksimalkan konversi autentikasi masuk dan pendaftaran akun pengguna dalam aplikasi (Firebase, 2022b).

Tabel 2.2 Dependensi FirebaseUI

|  |
| --- |
| dependencies {  // ...  implementation 'com.firebaseui:firebase-ui-auth:7.2.0'  } |

Sumber: Firebase (2022d)

#### 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, 2022e).

Tabel 2.3 Dependensi Firebase Cloud Firestore

|  |
| --- |
| dependencies {  // Impor BoM untuk *platform* Firebase  implementation *platform*('com.google.firebase: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.firebase:firebase-firestore'  } |

Sumber: Firebase (2022f)

#### 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, 2022g).

Tabel 2.4 Dependensi Firebase Cloud Storage

|  |
| --- |
| dependencies {  // Impor BoM untuk *platform* Firebase  implementation *platform*('com.google.firebase: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.firebase:firebase-storage-ktx'  } |

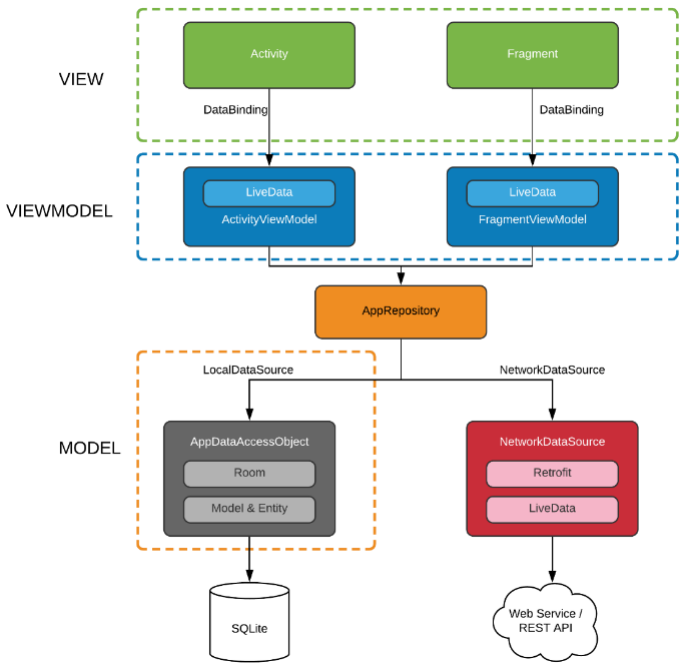
Sumber: Firebase (2022h)

## 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 memungkinakn 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 2.20 Rancangan arsitektur *MVVM*

Sumber: Arif, Musthafa, & Muriyatmoko (2019)

## 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).

### *Black Box Testing*

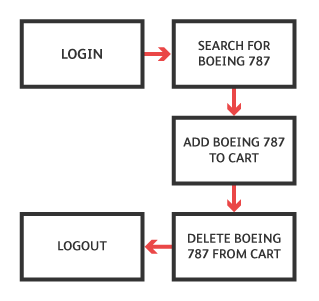
*Black box testing* adalah sebuah pendetakan pengujian dimana pengujian dilakukan dengan membuat kasus uji terhadap setiap fungsi. *Black box testing* digunakan untuk memastikan keselarasan semua alur perangkat lunak dengan kebutuhan pengguna. Pengujian ini dilakukan berdasarkan kebutuhan fungsional yang ada pada sistem (Muchlison, Kharisma, & Arwani, 2022).

### *Usability* *Testing*

*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).

### *Scenario-Based Testing*

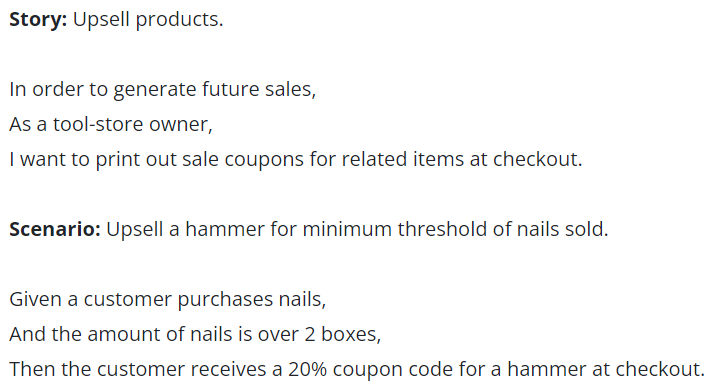
*Scenario* atau skenario adalah sebuah cerita yang bertujuan memecahkan tugas atau permasalahan yang rumit. Dalam pengujian, skenario adalah sebuah cara pengguna memanfaatkan perangkat lunak dalam kondisi dunia nyata guna memastikan verifikasi *end-to-end* fungsionalitas sistem. *Scenario-based testing* adalah sebuah pengujian yang berbasis skenario yang berfokus pada pengujian alur bisnis perangkat lunak. Tujuannya dalah 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 serangkian kasus-kasus pengujian yang saling terhubung (QATestLab, 2018).



Gambar 2.21 Contoh skenario pada *scenario-based testing*

Sumber: SoapUI (2022)

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



Gambar 2.22 Contoh sebuah *scenario-based testing*

Sumber: SoapUI (2022)

# METODOLOGI

## 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.

## 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

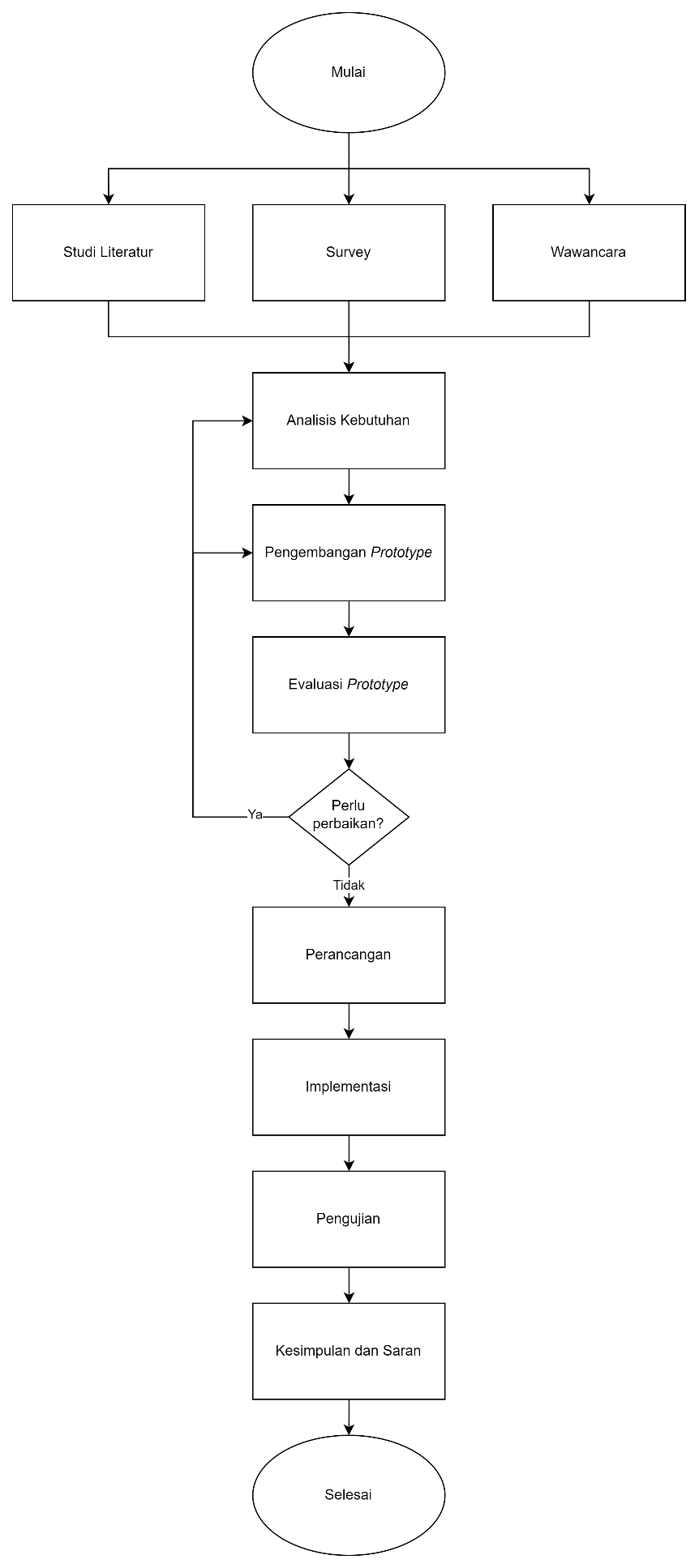
6. Implementasi

7. Pengujian

8. Kesimpulan dan Saran

9. Selesai

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 *Personas*, *Scenarios*, *User Stories*, dan *Features*. 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. TahapPerancangan 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.



Gambar 3.1 Metode penelitian

## 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.

## Lokasi Penelitian

Lokasi penelitian ini dilakukan di Fakultas Ilmu Komputer Universitas Brawijaya.

## 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 *Personas*, *User Stories*, *Scenarios*, dan *Features*. Pada tahap evaluasi *Prototype*, akan dilakukan pengumpulan data kembali secara daring dan luring menggunakan metode wawancara. Terakhir, akan dilakukan pengumpulan data untuk tahap pengujian dengan menggunakan survey via Google Form, dan juga dengan metode wawancara guna mendapatkan saran.

## 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 dengan sistem operasi Android 11.

## 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.

### Studi Literatur, Survey, Wawancara

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 *Personas, Scenarios, User Stories,* dan *Features*. Wawancara pada tahap Evaluasi *Prototype* digunakan untuk mengevaluasi *Prototype* yang telah dikembangkan.

### Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan mengembangkan *Personas, Scenarios, User Stories,* dan *Features,* yang didasari pada pengumpulan data yang bersumber dari wawancara. Analisis kebutuhan dengan pengembangan *Personas, Scenarios, User Stories,* dan *Features* bertujuan untuk mengetahui kebutuhan dan spesifikasi perangkat lunak yang ingin dibuat.

Berikut merupakan daftar pertanyaan wawancara kepada mashasiswa 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? |

### Pengembangan *Prototype*

Pengembangan *Prototype* dilakukan dengan menggunakan Figma, yang mengacu kepada kebutuhan dan spesifikasi perangkat lunak, yang mana diuraikan pada *Personas, Scenarios, User Stories,* dan *Features.* 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*.

### 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 *protoype* sesuai dengan perbaikan yang dibutuhkan. Evaluasi *Prototype* bertujuan untuk memvalidasi dan mengetahui apakah *Prototype* yang telah dibuat sudah sesuai dengan kebutuhan pengguna.

Berikut merupakan daftar pertanyaan wawancara kepada mashasiswa Fakultas Ilmu Komputer Universitas Brawijaya untuk Evaluasi *Prototype*.

Tabel 3.2 Daftar Pertanyaan Wawancara untuk Evaluasi *Prototype*

|  |  |
| --- | --- |
| **No** | **Pertanyaan** |
| 1 | Nama |
| 2 | Program Studi |
| 3 | Angkatan |
| 4 | Bagaimana hasil dari *Prototype* yang telah dicoba? Apakah cukup baik atau masih kurang baik? |
| 5 | Menurut anda, apakah terdapat fitur atau fungsionalitas yang masih kurang dari aplikasi ini? |
| 6 | Apakah ada kritik dan saran terhadap *Prototype* ini? |

### Perancangan

TahapPerancangan 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 algoritma dalam bentuk *pseudocode.* Perancangan bertujuan untuk membuat sebuah rancangan yang akan dipakai sebagai acuan dalam implementasi pengembangan perangkat lunak pada penelitian ini.

### 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. Implementasi bertujuan untuk menghasilkan sebuah perangkat lunak yang dapat digunakan oleh pengguna untuk mencapai tujuannya.

### Pengujian

Perangkat lunak yang telah dibuat selanjutnya akan diuji pada tahap Pengujian. Pengujian pada penelitian ini terdiri dari dua macam pengujian, yaitu *black box testing*, dan *usability testing*. *Black box testing* adalah sebuah pendetakan 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. Metode pengukuran *usability testing* yang digunakan pada penelitian ini yaitu *scenario-based testing* yang akan diuji kepada pengguna*. 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. Pengujian bertujuan untuk mengevaluasi kemampuan perangkat lunak, keselarasan perangkat lunak dengan kebutuhan pengguna, dan menentukan kesesuaian dengan hasil yang diharapkan.

### Kesimpulan dan Saran

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.

# ANALISIS KEBUTUHAN

**SUB BAB:**

* Pernyataan masalah yang lebih elaboratif/mendetail daripada yang di Pendahuluan.
* Identifikasi pemangku kepentingan (*stakeholders*) dan aktor (*actors*) sistem.

1. **Gambaran Umum Sistem(TIdak usah di desain solusi)**
2. **Identifikasi pengguna**
3. **Iterasi pertama**
   * + - 1. **Persona, user stories, scenario, features, prototype**
4. **Iterasi kedua**

Hasil berfungsi untuk melaporkan hasil pelaksanaan metode/teknik penelitian dan menyajikan data yang mendukung hasil tersebut. Penyajian data dan penjelasannya dilakukan secara terurut dan logis menggunakan teks dan ilustrasi lainnya (misalnya, tabel dan gambar). Urutan penjelasan dapat dilakukan secara kronologis berdasarkan urutan pelaksanaan metode atau berdasarkan tingkat kepentingan substansinya, dari yang lebih penting sampai ke yang proritasnya lebih rendah.

## Identifikasi Pengguna

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 4.1.

|  |  |
| --- | --- |
| 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. |

## *Prototype* Iterasi Pertama

Dalam pembuatan *prototype* iterasi pertama, digunakan analisis kebutuhan yang mengacu dari wawancara kepada pengguna. Pada tahap iterasi pertama dirancang *personas*, *scenarios*, *user stories*, *features*, dan *prototype* aplikasi dalam bentuk *high-fidelity*.

### *Personas*

*Personas* 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 menggunakannya, dan kesulitan yang mungkin akan dialami dalam memahami dan menggunakan fitur. Pada penelitian ini terdapat satu *Personas* yaitu Miko, seorang mahasiswa FILKOM UB.

|  |
| --- |
| **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 *smartphone*-nya untuk mencari resep di Google, dan video *tutorial* 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 tidak terlalu rumit baginya. |

### *Scenarios*

*Scenarios* 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 *scenarios* yang menggambarkan kegiatan memasak Miko.

|  |
| --- |
| **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. Untuk mengatasinya, ia mencari solusi dari permasalahan takaran dan resep di internet. Dari beberapa solusi, ia memilih untuk menggunakan sebuah aplikasi resep yang sederhana, dengan takaran yang jelas. Permasalahan lain yang ia hadapi adalah masalah harga dari bahan-bahan pada suatu resep. Ia harus menyesuaikan bahan yang terjangkau untuknya demi penghematan.  Ia mencoba mencari aplikasi resep sederhana dengan rincian harga perbahan dan bertanya kepada temannya Ziyad. 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. |

### *User Stories*(lebih ke as a role, penggunaan)

*User Stories* adalah sebuah deskripsi mengenai situasi dimana pengguna mencoba melakukan sesuatu dengan sistem perangkat lunak. Berikut merupakan sebuah *user stories* dari *scenarios* memasak nasi goreng. (banyak, detail, mencari resep masakan, sbg mahasiswa ingin tahu bahan bahan apa saja yang dibutuhkan, dan langkah-langkahnya, login dll semua fitur)->tabel isi paragraf

|  |  |  |
| --- | --- | --- |
| No | Kode *User Stories* | *User Stories* |
| 1. | T-1 | Sebagai Tamu, saya ingin dapat melakukan pendaftaran pada sistem sebagai Mahasiswa Kos. |
| 2. | T-2 | Sebagai Tamu, saya ingin masuk ke dalam sistem sebagai Mahasiswa Kos. |
| 3. | M-1 | Sebagai Mahasiswa Kos, saya ingin mencari resep masakan yang sesuai untuk mahasiswa, dimana resep tersebut sederhana dan tidak rumit, dengan takaran yang jelas. |
| 4. | M-2 | Sebagai Mahasiswa Kos, saya ingin mendapatkan rekomendasi resep masakan berdasarkan harga. |
| 5. | M-3 | Sebagai Mahasiswa Kos, saya ingin mengetahui bahan-bahan apa saja yang dibutuhkan dari suatu resep, harga dari tiap bahan, beserta langkah-langkah memasaknya. |

### *Features*(mendekati ke developer)

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian implementatif pengembangan sistem perangkat lunak.

Highlight terhadap *scenarios.*

|  |
| --- |
| **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 keinginanya 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. |

***Constraints***

Semua resep masakan yang disajikan haruslah sederhana, tidak rumit, menyertakan langkah dan takaran yang jelas.

***Description***

Resep masakan digunakan untuk memandu pengguna memasak makanan yang sederhana, dengan takaran yang jelas.

***Comments***

Untuk kedepannya, mungkin pengguna dapat menambahkan resep.

Fitur Resep Masakan

***Constraints***

Semua resep masakan yang direkomendasikan merupakan termasuk dari semua resep masakan yang ada di aplikasi dan bukan terpisah.

***Description***

Rekomendasi resep masakan adalah sebuah kategori daftar masakan tersendiri, yang akan merekomendasikan resep-resep pilihan.

***Comments***

Untuk kedepannya, mungkin pengguna dapat mengajukan rekomendasi resep.

Fitur Rekomendasi Resep

***Constraints***

Semua review terhadap resep yang ditambahkan harus dalam keadaan terautentikasi sebagai pengguna.

***Description***

Review resep adalah penilaian user terhadap sebuah resep, sehingga pengguna dapat mengetahui kualitas resep tersebut.

***Comments***

Untuk kedepannya, mungkin pengguna bisa mendapatkan feedback review yang diberikan.

Fitur Review Resep

## Iterasi Kedua

Contoh struktur skripsi untuk implementatif pembangunan dan nonimplementatif dapat dilihat pada kedua subbab berikut. (harga, waktu, estimasi masak lama atau tidak)

### Personas

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian implementatif pengembangan sistem perangkat lunak.

### Scenarios

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian implementatif pengembangan sistem perangkat lunak.

### User Stories

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian implementatif pengembangan sistem perangkat lunak.

### Features

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian implementatif pengembangan sistem perangkat lunak.

## Subbab Dua Satu

Sebelum menuliskan hasil ke dalam laporan, perlu dicermati dan ditentukan mana hasil yang relevan dan dapat digunakan untuk menjawab pertanyaan atau masalah penelitian. Hasil inilah yang perlu dimasukkan terlepas dari apakah hasil ini positif (misalnya, mendukung kebenaran hipotesis) atau negatif (misalnya, menolak hipotesis). Selanjutnya, perlu diperhatikan bagaimana menyajikannya dengan cara terbaik, apakah dengan teks, tabel atau gambar. Tabel dan gambar (foto, gambar, grafik, diagram) sering digunakan untuk mempresentasikan data yang detil dan kaya, sementara teks digunakan untuk menarasikan temuan yang lebih umum dan menjelaskan bagian-bagian tertentu yang menjadi fokus dalam tabel dan gambar.

## Subbab Dua Dua

Hasil dan pembahasan dapat diletakkan dengan kemungkinan berikut:

1. Dipisahkan secara fisik ke dalam bab-bab yang berbeda
2. Dipisahkan secara fisik ke dalam dua atau lebih paragraf atau subbab yang berbeda tetapi dalam bab yang sama
3. Dileburkan menjadi satu dalam paragraf, dijelaskan secara naratif-deskriptif, terdistribusi ke satu atau lebih bab yang ada

### Subbab Empat Dua Satu

Cara pertama atau kedua membantu pembaca yang ingin memisahkan observasi dan terjemahan dari observasi tersebut sehingga mereka dapat menilai kualitas dari masing-masing proses dengan lebih mudah. Kadang-kadang cara kedua lebih banyak dipilih daripada cara pertama jika data yang harus dipresentasikan yang cukup banyak dan laporan penelitian cukup panjang agar pembaca tidak perlu menunggu presentasi dari seluruh data selesai baru dapat membaca penerjemahannya. Cara pertama dan kedua ini banyak digunakan untuk penelitian yang bersifat kuantitatif, baik itu deskriptif, analitik, maupun implementatif.

### Subbab Empat Dua Dua

Cara ketiga biasanya digunakan jika data, analisis, dan penafsirannya sulit dipisahkan. Pemisahannya terkadang justru membuat laporan penelitian sulit dibaca. Hal ini dapat berlaku pada tipe penelitian yang bersifat kualitatif, baik itu deskriptif ataupun analitik/eksplanatori.

Pada dasarnya peletakan dan jumlah bab untuk hasil dan pembahasan sebaiknya disesuaikan karakter penelitian masing-masing. Judul bab pun tidak harus secara eksplisit “Hasil” dan “Pembahasan” tetapi dapat digantikan dengan nama yang lebih deskpritif dan tematik.

## Subbab Empat Tiga

Contoh struktur skripsi untuk implementatif pembangunan dan nonimplementatif dapat dilihat pada kedua subbab berikut.

### Contoh Struktur Penelitian Implementatif Pengembangan

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian implementatif pengembangan sistem perangkat lunak.

Bab 1 Pendahuluan

Bab 2 Landasan kepustakaan

Bab 3 Metodologi penelitian

Bab 4 Rekayasa persyaratan/kebutuhan

Bab 5 Perancangan dan implementasi

Bab 6 Pengujian

Bab 7 Penutup

Bab 1 sampai Bab 3 memuat informasi yang sesuai dengan panduan sebelumnya. Isi dari bab-bab berikutnya disesuaikan dengan syarat kecukupan skripsi untuk tipe implementatif berdasarkan keminatan masing-masing, seperti yang terdapat pada panduan kecukupan skripsi dan aturan khusus dari keminatan masing-masing. Di bawah ini adalah sebuah contoh saja:

* Bab 4 Persyaratan:
* Pernyataan masalah yang lebih elaboratif/mendetail daripada yang di Pendahuluan.
* Identifikasi pemangku kepentingan (*stakeholders*) dan aktor (*actors*) sistem.
* Daftar terstruktur persyaratan/kebutuhan perangkat lunak, secara fungsional, data, dan nonfungsional
* *Use cases*, *use case diagrams*, *use case specifications*, dan sebagainya.
* Bab 5 Perancangan dan implementasi:
* Rancangan arsitektur: deskripsi struktur dan setiap komponen utama
* Representasi data dalam model data dan basis data
* Detil implementasi dari fungsi-fungsi utama yang menjadi fokus
* Bab 6 Pengujian dan evaluasi
* Strategi, rencana, kasus, dan data pengujian
* Ringkasan hasil pengujian perangkat lunak, termasuk data dan analisisnya (detilnya di Lampiran)
* Evaluasi hasil proyek secara keseluruhan, misalkan
* Bab 7 Penutup
* Ringkasan dari capaian proyek
* Saran pengembangan lebih lanjut

Pada contoh struktur ini “hasil” tersebar di beberapa bab mulai Bab 4 Persyaratan sampai Bab 6, sedangkan “pembahasan” secara keseluruhan terhadap masalah penelitian terdapat di Bab 6. Yang dimaksud dengan pengujian dalam Bab 6 terfokus pada pengujian persyaratan perangkat lunak, sedangkan evaluasi berfungsi sebagai “pembahasan” secara keseluruhan, yaitu menentukan apakah “hasil” sudah menjawab masalah penelitian yang dirumuskan pada Bab 1.

Sebagai catatan, Bab 3 Metodologi Penelitian umumnya menjelaskan model proses perangkat lunak yang digunakan. Jika strategi untuk setiap aktivitasnya (analisis persyaratan, perancangan, dan seterusnya) sudah dijelaskan di Bab 3 ini juga, maka bab-bab lainnya yang berhubungan dengan aktivitas-aktivitas ini masing-masing langsung dapat menjelaskan hasil pelaksanaan metodenya.

### Contoh Struktur Penelitian Nonimplementatif

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian nonimplementatif.

Bab 1 Pendahuluan

Bab 2 Landasan kepustakaan

Bab 3 Metodologi Penelitian

Bab 4 Hasil

Bab 5 Pembahasan

Bab 6 Penutup

Isi dari setiap bab dapat menyesuaikan dengan panduan yang telah dijelaskan sebelumnya. Jika diperlukan, Bab 4 dapat digabungkan dengan Bab 5, menjadi Hasil dan Pembahasan.

Struktur dasar ini cukup universal sehingga dapat digunakan juga untuk tipe-tipe penelitian lainnya, khususnya jika belum ada struktur lain yang lebih tematik dan cocok untuk penelitian yang bersangkutan. Untuk lebih tepatnya, struktur penulisan menyesuaikan dengan studi keminatan serta saran dari dosen pembimbing masing-masing.

# HASIL

Hasil berfungsi untuk melaporkan hasil pelaksanaan metode/teknik penelitian dan menyajikan data yang mendukung hasil tersebut. Penyajian data dan penjelasannya dilakukan secara terurut dan logis menggunakan teks dan ilustrasi lainnya (misalnya, tabel dan gambar). Urutan penjelasan dapat dilakukan secara kronologis berdasarkan urutan pelaksanaan metode atau berdasarkan tingkat kepentingan substansinya, dari yang lebih penting sampai ke yang proritasnya lebih rendah.

## Subbab Dua Satu

Sebelum menuliskan hasil ke dalam laporan, perlu dicermati dan ditentukan mana hasil yang relevan dan dapat digunakan untuk menjawab pertanyaan atau masalah penelitian. Hasil inilah yang perlu dimasukkan terlepas dari apakah hasil ini positif (misalnya, mendukung kebenaran hipotesis) atau negatif (misalnya, menolak hipotesis). Selanjutnya, perlu diperhatikan bagaimana menyajikannya dengan cara terbaik, apakah dengan teks, tabel atau gambar. Tabel dan gambar (foto, gambar, grafik, diagram) sering digunakan untuk mempresentasikan data yang detil dan kaya, sementara teks digunakan untuk menarasikan temuan yang lebih umum dan menjelaskan bagian-bagian tertentu yang menjadi fokus dalam tabel dan gambar.

## Subbab Dua Dua

Hasil dan pembahasan dapat diletakkan dengan kemungkinan berikut:

1. Dipisahkan secara fisik ke dalam bab-bab yang berbeda
2. Dipisahkan secara fisik ke dalam dua atau lebih paragraf atau subbab yang berbeda tetapi dalam bab yang sama
3. Dileburkan menjadi satu dalam paragraf, dijelaskan secara naratif-deskriptif, terdistribusi ke satu atau lebih bab yang ada

### Subbab Empat Dua Satu

Cara pertama atau kedua membantu pembaca yang ingin memisahkan observasi dan terjemahan dari observasi tersebut sehingga mereka dapat menilai kualitas dari masing-masing proses dengan lebih mudah. Kadang-kadang cara kedua lebih banyak dipilih daripada cara pertama jika data yang harus dipresentasikan yang cukup banyak dan laporan penelitian cukup panjang agar pembaca tidak perlu menunggu presentasi dari seluruh data selesai baru dapat membaca penerjemahannya. Cara pertama dan kedua ini banyak digunakan untuk penelitian yang bersifat kuantitatif, baik itu deskriptif, analitik, maupun implementatif.

### Subbab Empat Dua Dua

Cara ketiga biasanya digunakan jika data, analisis, dan penafsirannya sulit dipisahkan. Pemisahannya terkadang justru membuat laporan penelitian sulit dibaca. Hal ini dapat berlaku pada tipe penelitian yang bersifat kualitatif, baik itu deskriptif ataupun analitik/eksplanatori.

Pada dasarnya peletakan dan jumlah bab untuk hasil dan pembahasan sebaiknya disesuaikan karakter penelitian masing-masing. Judul bab pun tidak harus secara eksplisit “Hasil” dan “Pembahasan” tetapi dapat digantikan dengan nama yang lebih deskpritif dan tematik.

## Subbab Empat Tiga

Contoh struktur skripsi untuk implementatif pembangunan dan nonimplementatif dapat dilihat pada kedua subbab berikut.

### Contoh Struktur Penelitian Implementatif Pengembangan

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian implementatif pengembangan sistem perangkat lunak.

Bab 1 Pendahuluan

Bab 2 Landasan kepustakaan

Bab 3 Metodologi penelitian

Bab 4 Rekayasa persyaratan/kebutuhan

Bab 5 Perancangan dan implementasi

Bab 6 Pengujian

Bab 7 Penutup

Bab 1 sampai Bab 3 memuat informasi yang sesuai dengan panduan sebelumnya. Isi dari bab-bab berikutnya disesuaikan dengan syarat kecukupan skripsi untuk tipe implementatif berdasarkan keminatan masing-masing, seperti yang terdapat pada panduan kecukupan skripsi dan aturan khusus dari keminatan masing-masing. Di bawah ini adalah sebuah contoh saja:

* Bab 4 Persyaratan:
* Pernyataan masalah yang lebih elaboratif/mendetail daripada yang di Pendahuluan.
* Identifikasi pemangku kepentingan (*stakeholders*) dan aktor (*actors*) sistem.
* Daftar terstruktur persyaratan/kebutuhan perangkat lunak, secara fungsional, data, dan nonfungsional
* *Use cases*, *use case diagrams*, *use case specifications*, dan sebagainya.
* Bab 5 Perancangan dan implementasi:
* Rancangan arsitektur: deskripsi struktur dan setiap komponen utama
* Representasi data dalam model data dan basis data
* Detil implementasi dari fungsi-fungsi utama yang menjadi fokus
* Bab 6 Pengujian dan evaluasi
* Strategi, rencana, kasus, dan data pengujian
* Ringkasan hasil pengujian perangkat lunak, termasuk data dan analisisnya (detilnya di Lampiran)
* Evaluasi hasil proyek secara keseluruhan, misalkan
* Bab 7 Penutup
* Ringkasan dari capaian proyek
* Saran pengembangan lebih lanjut

Pada contoh struktur ini “hasil” tersebar di beberapa bab mulai Bab 4 Persyaratan sampai Bab 6, sedangkan “pembahasan” secara keseluruhan terhadap masalah penelitian terdapat di Bab 6. Yang dimaksud dengan pengujian dalam Bab 6 terfokus pada pengujian persyaratan perangkat lunak, sedangkan evaluasi berfungsi sebagai “pembahasan” secara keseluruhan, yaitu menentukan apakah “hasil” sudah menjawab masalah penelitian yang dirumuskan pada Bab 1.

Sebagai catatan, Bab 3 Metodologi Penelitian umumnya menjelaskan model proses perangkat lunak yang digunakan. Jika strategi untuk setiap aktivitasnya (analisis persyaratan, perancangan, dan seterusnya) sudah dijelaskan di Bab 3 ini juga, maka bab-bab lainnya yang berhubungan dengan aktivitas-aktivitas ini masing-masing langsung dapat menjelaskan hasil pelaksanaan metodenya.

### Contoh Struktur Penelitian Nonimplementatif

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian nonimplementatif.

Bab 1 Pendahuluan

Bab 2 Landasan kepustakaan

Bab 3 Metodologi Penelitian

Bab 4 Hasil

Bab 5 Pembahasan

Bab 6 Penutup

Isi dari setiap bab dapat menyesuaikan dengan panduan yang telah dijelaskan sebelumnya. Jika diperlukan, Bab 4 dapat digabungkan dengan Bab 5, menjadi Hasil dan Pembahasan.

Struktur dasar ini cukup universal sehingga dapat digunakan juga untuk tipe-tipe penelitian lainnya, khususnya jika belum ada struktur lain yang lebih tematik dan cocok untuk penelitian yang bersangkutan. Untuk lebih tepatnya, struktur penulisan menyesuaikan dengan studi keminatan serta saran dari dosen pembimbing masing-masing.

# HASIL

Hasil berfungsi untuk melaporkan hasil pelaksanaan metode/teknik penelitian dan menyajikan data yang mendukung hasil tersebut. Penyajian data dan penjelasannya dilakukan secara terurut dan logis menggunakan teks dan ilustrasi lainnya (misalnya, tabel dan gambar). Urutan penjelasan dapat dilakukan secara kronologis berdasarkan urutan pelaksanaan metode atau berdasarkan tingkat kepentingan substansinya, dari yang lebih penting sampai ke yang proritasnya lebih rendah.

## Subbab Dua Satu

Sebelum menuliskan hasil ke dalam laporan, perlu dicermati dan ditentukan mana hasil yang relevan dan dapat digunakan untuk menjawab pertanyaan atau masalah penelitian. Hasil inilah yang perlu dimasukkan terlepas dari apakah hasil ini positif (misalnya, mendukung kebenaran hipotesis) atau negatif (misalnya, menolak hipotesis). Selanjutnya, perlu diperhatikan bagaimana menyajikannya dengan cara terbaik, apakah dengan teks, tabel atau gambar. Tabel dan gambar (foto, gambar, grafik, diagram) sering digunakan untuk mempresentasikan data yang detil dan kaya, sementara teks digunakan untuk menarasikan temuan yang lebih umum dan menjelaskan bagian-bagian tertentu yang menjadi fokus dalam tabel dan gambar.

## Subbab Dua Dua

Hasil dan pembahasan dapat diletakkan dengan kemungkinan berikut:

1. Dipisahkan secara fisik ke dalam bab-bab yang berbeda
2. Dipisahkan secara fisik ke dalam dua atau lebih paragraf atau subbab yang berbeda tetapi dalam bab yang sama
3. Dileburkan menjadi satu dalam paragraf, dijelaskan secara naratif-deskriptif, terdistribusi ke satu atau lebih bab yang ada

### Subbab Empat Dua Satu

Cara pertama atau kedua membantu pembaca yang ingin memisahkan observasi dan terjemahan dari observasi tersebut sehingga mereka dapat menilai kualitas dari masing-masing proses dengan lebih mudah. Kadang-kadang cara kedua lebih banyak dipilih daripada cara pertama jika data yang harus dipresentasikan yang cukup banyak dan laporan penelitian cukup panjang agar pembaca tidak perlu menunggu presentasi dari seluruh data selesai baru dapat membaca penerjemahannya. Cara pertama dan kedua ini banyak digunakan untuk penelitian yang bersifat kuantitatif, baik itu deskriptif, analitik, maupun implementatif.

### Subbab Empat Dua Dua

Cara ketiga biasanya digunakan jika data, analisis, dan penafsirannya sulit dipisahkan. Pemisahannya terkadang justru membuat laporan penelitian sulit dibaca. Hal ini dapat berlaku pada tipe penelitian yang bersifat kualitatif, baik itu deskriptif ataupun analitik/eksplanatori.

Pada dasarnya peletakan dan jumlah bab untuk hasil dan pembahasan sebaiknya disesuaikan karakter penelitian masing-masing. Judul bab pun tidak harus secara eksplisit “Hasil” dan “Pembahasan” tetapi dapat digantikan dengan nama yang lebih deskpritif dan tematik.

## Subbab Empat Tiga

Contoh struktur skripsi untuk implementatif pembangunan dan nonimplementatif dapat dilihat pada kedua subbab berikut.

### Contoh Struktur Penelitian Implementatif Pengembangan

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian implementatif pengembangan sistem perangkat lunak.

Bab 1 Pendahuluan

Bab 2 Landasan kepustakaan

Bab 3 Metodologi penelitian

Bab 4 Rekayasa persyaratan/kebutuhan

Bab 5 Perancangan dan implementasi

Bab 6 Pengujian

Bab 7 Penutup

Bab 1 sampai Bab 3 memuat informasi yang sesuai dengan panduan sebelumnya. Isi dari bab-bab berikutnya disesuaikan dengan syarat kecukupan skripsi untuk tipe implementatif berdasarkan keminatan masing-masing, seperti yang terdapat pada panduan kecukupan skripsi dan aturan khusus dari keminatan masing-masing. Di bawah ini adalah sebuah contoh saja:

* Bab 4 Persyaratan:
* Pernyataan masalah yang lebih elaboratif/mendetail daripada yang di Pendahuluan.
* Identifikasi pemangku kepentingan (*stakeholders*) dan aktor (*actors*) sistem.
* Daftar terstruktur persyaratan/kebutuhan perangkat lunak, secara fungsional, data, dan nonfungsional
* *Use cases*, *use case diagrams*, *use case specifications*, dan sebagainya.
* Bab 5 Perancangan dan implementasi:
* Rancangan arsitektur: deskripsi struktur dan setiap komponen utama
* Representasi data dalam model data dan basis data
* Detil implementasi dari fungsi-fungsi utama yang menjadi fokus
* Bab 6 Pengujian dan evaluasi
* Strategi, rencana, kasus, dan data pengujian
* Ringkasan hasil pengujian perangkat lunak, termasuk data dan analisisnya (detilnya di Lampiran)
* Evaluasi hasil proyek secara keseluruhan, misalkan
* Bab 7 Penutup
* Ringkasan dari capaian proyek
* Saran pengembangan lebih lanjut

Pada contoh struktur ini “hasil” tersebar di beberapa bab mulai Bab 4 Persyaratan sampai Bab 6, sedangkan “pembahasan” secara keseluruhan terhadap masalah penelitian terdapat di Bab 6. Yang dimaksud dengan pengujian dalam Bab 6 terfokus pada pengujian persyaratan perangkat lunak, sedangkan evaluasi berfungsi sebagai “pembahasan” secara keseluruhan, yaitu menentukan apakah “hasil” sudah menjawab masalah penelitian yang dirumuskan pada Bab 1.

Sebagai catatan, Bab 3 Metodologi Penelitian umumnya menjelaskan model proses perangkat lunak yang digunakan. Jika strategi untuk setiap aktivitasnya (analisis persyaratan, perancangan, dan seterusnya) sudah dijelaskan di Bab 3 ini juga, maka bab-bab lainnya yang berhubungan dengan aktivitas-aktivitas ini masing-masing langsung dapat menjelaskan hasil pelaksanaan metodenya.

### Contoh Struktur Penelitian Nonimplementatif

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian nonimplementatif.

Bab 1 Pendahuluan

Bab 2 Landasan kepustakaan

Bab 3 Metodologi Penelitian

Bab 4 Hasil

Bab 5 Pembahasan

Bab 6 Penutup

Isi dari setiap bab dapat menyesuaikan dengan panduan yang telah dijelaskan sebelumnya. Jika diperlukan, Bab 4 dapat digabungkan dengan Bab 5, menjadi Hasil dan Pembahasan.

Struktur dasar ini cukup universal sehingga dapat digunakan juga untuk tipe-tipe penelitian lainnya, khususnya jika belum ada struktur lain yang lebih tematik dan cocok untuk penelitian yang bersangkutan. Untuk lebih tepatnya, struktur penulisan menyesuaikan dengan studi keminatan serta saran dari dosen pembimbing masing-masing.

# PEMBAHASAN

Pembahasan berfungsi untuk menerjemahkan makna dari hasil yang diperoleh untuk menjawab pertanyaan atau masalah penelitian. Fungsi lainnya adalah untuk menjelaskan pemahaman baru yang didapatkan dari hasil penelitian, yang diharapkan berguna dalam pengembangan keilmuan. Dalam penelitian tingkat lanjut, fungsi pembahasan yang kedua ini sangat penting karena dapat menunjukkan kontribusi penulis terhadap pengembangan keilmuan. Akan tetapi, dalam penelitian tingkat skripsi, fungsi yang kedua ini dapat diterapkan secara terbatas karena pendidikan S1 tidak dituntut untuk pengembangan keilmuan secara substansial, tetapi cukup terhadap pemahaman personal dalam implementasi konsep atau teori.

## Subbab Lima Satu

Dalam menjawab masalah penelitian, penulis diminta untuk melakukan evaluasi kritis terhadap hasil yang diperoleh. Tergantung dari fokus penelitian, beberapa contoh pertanyaan kritis yang dapat dijawab adalah:

* Seberapa jauh tujuan penelitian telah tercapai?
* Apakah aplikasi atau sistem yang dibangun sesuai dengan tujuannya?
* Apakah metode atau praktik perancangan dan implementasi yang baik telah dijalankan?
* Apakah teknologi implementasi yang tepat telah dipilih? Dan sebagainya.

### Subbab Lima Satu Satu

Dalam menjelaskan pemahaman baru yang didapatkan, penulis dapat mengubungkan hasil penelitian dengan pengetahuan teoritik atau penelitian sebelumnya yang telah dibahas. Kaitan antara hasil penelitian dan pengetahuan teoritik misalnya berupa:

* pendapat tentang metode yang digunakan dari pustaka, apakah dapat digunakan dengan baik secara langsung, dengan penyesuaian, atau dengan batasan tertentu;
* konfirmasi tentang batasan dari metodologi yang digunakan sehingga dapat berpengaruh pada hasil;
* penjelasan tentang informasi penting pada penelitian lainnya yang membantu penulis untuk menerjemahkan data penelitian penulis;
* penjelasan tentang kemungkinan hasil dari penelitian lainnya yang dapat dikombinasikan dengan penelitian penulis untuk memberikan pengetahuan baru; dan sebagainya.

### Subbab Lima Satu Dua

Penulis dapat merefleksikan apa yang telah dipelajari selama melakukan penelitian, tetapi harus tetap terfokus dengan masalah penelitian ini dan tidak melebar ke masalah lainnya. Hal-hal yang berada di luar fokus peneltian tetapi penting dan menarik untuk diteliti dapat disarankan sebagai bahan penelitian berikutnya. Hal ini dapat dipertegas di bab Kesimpulan/ Penutup.

## Subbab Lima Dua

Hasil dan pembahasan dapat diletakkan dengan kemungkinan berikut:

1. Dipisahkan secara fisik ke dalam bab-bab yang berbeda
2. Dipisahkan secara fisik ke dalam dua atau lebih paragraf atau subbab yang berbeda tetapi dalam bab yang sama
3. Dileburkan menjadi satu dalam paragraf, dijelaskan secara naratif-deskriptif, terdistribusi ke satu atau lebih bab yang ada

### Subbab Lima Dua Satu

Cara pertama atau kedua membantu pembaca yang ingin memisahkan observasi dan terjemahan dari observasi tersebut sehingga mereka dapat menilai kualitas dari masing-masing proses dengan lebih mudah. Kadang-kadang cara kedua lebih banyak dipilih daripada cara pertama jika data yang harus dipresentasikan yang cukup banyak dan laporan penelitian cukup panjang agar pembaca tidak perlu menunggu presentasi dari seluruh data selesai baru dapat membaca penerjemahannya. Cara pertama dan kedua ini banyak digunakan untuk penelitian yang bersifat kuantitatif, baik itu deskriptif, eksplanatori, maupun implementatif.

### Subbab Lima Dua Dua

Cara ketiga biasanya digunakan jika data, analisis, dan penafsirannya sulit dipisahkan. Pemisahannya terkadang justru membuat laporan penelitian sulit dibaca. Hal ini dapat berlaku pada tipe penelitian yang bersifat kualitatif, baik itu deskriptif ataupun analitik/eksplanatori.

Pada dasarnya peletakan dan jumlah bab untuk hasil dan pembahasan sebaiknya disesuaikan karakter penelitian masing-masing. Judul bab pun tidak harus secara eksplisit “Hasil” dan “Pembahasan” tetapi dapat digantikan dengan nama yang lebih deskpritif dan tematik.

## Subbab Lima Tiga

Contoh struktur skripsi untuk implementatif pembangunan dan nonimplementatif eksperimental dapat dilihat pada kedua subbab berikut.

### Contoh Struktur Penelitian Implementatif Pembangunan

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian implementatif pembangunan sistem perangkat lunak.

Bab 1 Pendahuluan

Bab 2 Landasan Kepustakaan

Bab 3 Metodologi Penelitian

Bab 4 Persyaratan

Bab 5 Perancangan dan Implementasi

Bab 6 Pengujian

Bab 7 Penutup

Bab 1 sampai Bab 3 memuat informasi yang sesuai dengan panduan sebelumnya. Isi dari bab-bab berikutnya:

* Bab 4 Persyaratan:
* Pernyataan masalah (problem statement), yang lebih elaboratif daripada yang di Pendahuluan.
* Identifikasi pemangku kepentingan (stakeholders) dan aktor (actors) sistem.
* Daftar terstruktur persyaratan/kebutuhan perangkat lunak, secara fungsional, data, dan non-fungsional
* Use cases, use case diagrams, dan use case specifications, dan sebagainya.
* Bab 5 Perancangan dan Implementasi:
* Rancangan arsitektur: deskripsi struktur dan setiap komponen utama
* Representasi data dalam model data dan basis data
* Detil implementasi dari fungsi-fungsi utama yang menjadi fokus
* Bab 6 Pengujian dan Evaluasi
* Strategi, rencana, kasus, dan data pengujian
* Ringkasan hasil pengujian perangkat lunak, termasuk data dan analisisnya (detilnya di Lampiran)
* Evaluasi hasil proyek secara keseluruhan, misalkan
* Bab 7 Penutup
* Ringkasan dari capaian proyek
* Saran pengembangan lebih lanjut

Pada contoh struktur ini “hasil” tersebar di beberapa bab mulai Bab 4 Persyaratan sampai Bab 6, sedangkan “pembahasan” secara keseluruhan terhadap masalah penelitian terdapat di Bab 6. Yang dimaksud dengan pengujian dalam Bab 6 terfokus pada pengujian persyaratan perangkat lunak, sedangkan evaluasi berfungsi sebagai “pembahasan” secara keseluruhan, yaitu menentukan apakah “hasil” sudah menjawab masalah penelitian yang dirumuskan pada Bab 1.

Sebagai catatan, Bab 3 Metodologi umumnya menjelaskan model proses perangkat lunak yang digunakan. Jika strategi untuk setiap aktivitasnya (analisis persyaratan, perancangan, dan seterusnya) sudah dijelaskan di Bab 3 ini juga, maka bab-bab lainnya yang berhubungan dengan aktivitas-aktivitas ini masing-masing langsung dapat menjelaskan hasil pelaksanaan metodenya.

### Contoh Struktur Penelitian Nonimplementatif Eksperimental

Berikut ini adalah contoh bab-bab yang terdapat pada penelitian implementatif pembangunan sistem perangkat lunak.

Bab 1 Pendahuluan

Bab 2 Landasan Kepustakaan

Bab 3 Metodologi Penelitian

Bab 4 Hasil

Bab 5 Pembahasan

Bab 6 Penutup

Isi dari setiap bab dapat menyesuaikan dengan panduan yang telah dijelaskan sebelumnya. Jika diperlukan, Bab 4 dapat digabungkan dengan Bab 5, menjadi Hasil dan Pembahasan.

Struktur dasar ini cukup universal sehingga dapat digunakan juga untuk tipe-tipe penelitian lainnya, khususnya jika belum ada struktur lain yang lebih tematik dan cocok untuk penelitian yang bersangkutan.

# Penutup

Bagian ini memuat kesimpulan dan saran terhadap skripsi. Kesimpulan dan saran disajikan secara terpisah, dengan penjelasan sebagai berikut:

## Kesimpulan

Kesimpulan merupakan pernyataan-pernyataan yang singkat, jelas, dan tepat tentang hasil penelitian yang diperoleh berdasarkan tujuannya. Bagian ini merupakan penegasan dari yang telah dijelaskan pada bagian Pembahasan dan tidak memuat informasi yang baru. Bagian ini juga mencerminkan jawaban dari rumusan masalah (pertanyaan penelitian).

## Saran

Saran berisi pernyataan-pernyataan yang ringkas dan jelas tentang masalah-masalah atau hal-hal yang dapat dilakukan untuk mengembangkan penelitian ini lebih lanjut. Saran itu dapat diarahkan pada aspek metode, instrumen, populasi/sampel, dan sebagainya.

DAFTAR REFERENSI

Yasuma, A., 2017. *Resep Masakan Khas Lombok Berbasis Android*. Diploma. Sekolah Tinggi Manajemen Informatika Dan Komputer Akakom. Tersedia di <<https://eprints.utdi.ac.id/4908/>> [Diakses 26 Juli 2022]

Karlina, L., Asian, J., & Mahmud, M., 2019. Rancang Bangun Aplikasi Resep Masakan Menggunakan Metode Xp Berbasis Android. *Jurnal Rekayasa Teknologi Nusa Putra,* [e-journal] 08(01), 18*.* Tersedia melalui: Perreferensian Universitas Nusa Putra <<https://jurnal.nusaputra.ac.id/rekayasa/uploads/paper/2e89e-jurnal-skripsi-finish-lina.pdf>> [Diakses 26 Juli 2022]

Santoso, T. A., 2016. *Aplikasi Pencarian Resep Masakan Berbasis Mobile Web Berdasarkan Ketersediaan Bahan Dengan Metode Simple Additive Weighting.* S1. Universitas Islam Indonesia Yogyakarta. Tersedia di <<https://dspace.uii.ac.id/bitstream/handle/123456789/3781/04%20abstract.pdf?sequence=5&isAllowed=y>> [Diakses 26 Juli 2022]

Agustina, R., & Suprianto, D., 2012. *Pemrograman Aplikasi Android.* [e-book] ResearchGate. Tersedia melalui: Website ResearchGate <<https://www.researchgate.net/publication/338819838_Pemrograman_Aplikasi_Android>>[Diakses 26 Juli 2022]

Ichwan, B. N., Sujalwo, & Supardi, A., 2013. *Perancangan Aplikasi Resep Masakan Khas Jawa Berbasis Android.* S2. Universitas Muhammadiyah Surakarta. Tersedia di <<http://eprints.ums.ac.id/24179/>> [Diakses 26 Juli 2022]

Statcounter, 2022. *Mobile Operating System Market Share Indonesia | Statcounter Global Stats*. [online] Tersedia di: <<https://gs.statcounter.com/os-market-share/mobile/indonesia>> [Diakses 26 Juli 2022]

Google Forms, 2022. *Survey Aplikasi Resep Masakan (Responses) – Google Sheets*. [online] Tersedia di: <<https://docs.google.com/spreadsheets/d/141gI535YMwryl7sANbOwdk1XW6grMyn6G1Bad3bnOwk/edit?usp=sharing>> [Diakses 27 Juli 2022]

Google Forms, 2022. *Survey Aplikasi Panduan dan Resep Masakan Mahasiswa Filkom Universitas Brawijaya (Responses)*. [online] Tersedia di: <<https://docs.google.com/spreadsheets/d/141gI535YMwryl7sANbOwdk1XW6grMyn6G1Bad3bnOwk/edit?usp=sharing>> [Diakses 24 Agustus 2022]

Muchlison, I. D., Kharisma, A. P., & Arwani, I., 2022. Pengembangan Aplikasi Perangkat Bergerak Sistem Informasi Event di bidang Teknologi Informasi berbasis Android. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* [e-journal] 6(1), 282-291. Tersedia melalui: Perreferensian Universitas Brawijaya <<https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/10464/4642>> [Diakses 8 Agustus 2022]

Arifien, A., 2022. *Perancangan Buku Ilustrasi Resep Makanan Sehat Sebagai Inspirasi Memulai Bisnis Di Sektor Kuliner.* S1. Universitas Pembangunan Nasional “Veteran” Jawa Timur. Tersedia di <<http://repository.upnjatim.ac.id/6529/>> [Diakses 8 Agustus 2022]

Priyanto, R., 2010. *Analisis Kebutuhan Perlengkapan Bengkel Otomotif Smk Swasta di Karanganyar.* S1. Universitas Sebelas Maret Surakarta. Tersedia di <<https://123dok.com/document/1y9vkvwq-analisis-kebutuhan-perlengkapan-bengkel-otomotif-smk-swasta-karanganyar.html>> [Diakses 8 Agustus 2022]

Muradi, 2021. *Penerapan Metode Earliest Due Date pada Layanan Homecare Kesehatan Ibu dan Anak Berbasis Android pada Desa Penengahan Pesisir Barat.* S1. Institut Informatika dan Bisnis Darmajaya Bandar Lampung. Tersedia di <<http://repo.darmajaya.ac.id/6064/>> [Diakses 8 Agustus 2022]

Abdelbaki, I., Charkaoui, S., Lahmar, E. H. B., & Marzak, A., 2016. Cross-*platform* Mobile Development based on MDA Approach. *International Journal of Interactive Mobile Technologies (iJIM),* [e-journal] 10(4), 18-25. Tersedia melalui: Perreferensial jurnal iJIM <<https://online-journals.org/index.php/i-jim/article/view/5570>> [Diakses 8 Agustus 2022]

Putra, I. D. G. D., 2017. *Aplikasi Pembelajaran Biologi untuk Siswa Kelas XII Sekolah Menengah Atas Berbasis Android.* S1. Sekolah Tinggi Manajeman Informatika dan Komputer Akakom Yogyakarta. Tersedia di <<https://eprints.utdi.ac.id/4897/>> [Diakses 8 Agustus 2022]

Nugraha, C., 2019. *Implementasi Fitur pada Firebase untuk Aplikasi Chat Messagers Berbasis Android.* S1. Sekolah Tinggi Manajeman Informatika dan Komputer Akakom Yogyakarta. Tersedia di <<https://eprints.utdi.ac.id/8733/>> [Diakses 9 Agustus 2022]

Firebase, 2022a. *Firebase Products*. [online] Tersedia di: <<https://firebase.google.com/products-build>> [Diakses 9 Agustus 2022].

Firebase, 2022b. *Firebase Authentication | Simple, multi-platform sign-in*. [online] Tersedia di: <<https://firebase.google.com/products/auth>> [Diakses 9 Agustus 2022].

Firebase, 2022c. *Get Started with Firebase Authentication on Android*. [online] Tersedia di: <<https://firebase.google.com/docs/auth/android/start>> [Diakses 9 Agustus 2022].

Firebase, 2022d. *Easily add sign-in to your Android app with FirebaseUI*. [online] Tersedia di: <<https://firebase.google.com/docs/auth/android/firebaseui>> [Diakses 9 Agustus 2022].

Firebase, 2022e. *Cloud Firestore | Store and sync app data at global scale | Firebase*. [online] Tersedia di: <<https://firebase.google.com/products/firestore>> [Diakses 9 Agustus 2022].

Firebase, 2022f. *Get started with Cloud Firestore | Firebase*. [online] Tersedia di: <<https://firebase.google.com/docs/firestore/quickstart>> [Diakses 9 Agustus 2022].

Firebase, 2022g. *Get started with Cloud Firestore | Firebase*. [online] Tersedia di: <<https://firebase.google.com/docs/firestore/quickstart>> [Diakses 9 Agustus 2022].

Firebase, 2022h. *Get started with Cloud Storage on Android | Firebase Storage*. [online] Tersedia di: <<https://firebase.google.com/docs/storage/android/start>> [Diakses 9 Agustus 2022].

Arif, M. S., Musthafa, A., & Muriyatmoko, D., 2019. *Implementasi Pola Arsitektur Model-View-ViewModel (MVVM) pada Sistem Informasi Akademik Universitas Darussalam Gontor Berbasis Mobile.* [pdf] ResearchGate. Tersedia di: <[https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775\_Implementasi\_Pola\_Arsitektur\_](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[Model](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[-](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[View](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[-](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[ViewModel](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[\_](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[MVVM](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[\_pada\_Sistem\_Informasi\_Akademik\_Universitas\_Darussalam\_Gontor\_Berbasis\_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[Model](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[-](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[View](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[-](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[ViewModel](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[-](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[MVVM](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)*[-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf](https://www.researchgate.net/profile/Muhammad-Syakir-Arif/publication/339616775_Implementasi_Pola_Arsitektur_Model-View-ViewModel_MVVM_pada_Sistem_Informasi_Akademik_Universitas_Darussalam_Gontor_Berbasis_Mobile/links/5e5c9f5b92851cefa1d4b43a/Implementasi-Pola-Arsitektur-Model-View-ViewModel-MVVM-pada-Sistem-Informasi-Akademik-Universitas-Darussalam-Gontor-Berbasis-Mobile.pdf)> [Diakses 9 Agustus 2022].

Raj, P., Raman, A., & Subramanian, H., 2017. *Architectural Patterns*. Birmingham: Packt Publisher.

Academia, 2022. *Metode Pengembangan Perangkat Lunak.* [pdf] Academia. Tersedia di: <<https://www.academia.edu/4844015/Metode_pengembangan_perangkat_lunak>> [Diakses 10 Agustus 2022]

Aisah, S., Irfansyahfalah, M., Kusyadi, I., Noviyanto, D., Saifudin, A., & Setiany, A. P., 2021. Penggunaan Metode System Development Life Cycle ( *SDLC*) dalam Analasis dan Perancangan Sistem Informasi Penerimaan Kas Sekolah. *Jurnal Teknologi Sistem Informasi dan Aplikasi,* [e-journal]4(3), 179-186. Tersedia melalui: Perreferensian Universitas Pamulang <<http://openjournal.unpam.ac.id/index.php/JTSI/article/view/11992/pdf>> [Diakses 10 Agustus 2022]

Dahono, G. B. S., 2020. *Pengujian Black Box Menggunakan Metode Decision Table Testing Pada Google Speech-To-Text.* S1. Sekolah Tinggi Manajeman Informatika dan Komputer Akakom Yogyakarta. Tersedia di <<https://eprints.utdi.ac.id/9083/>> [Diakses 10 Agustus 2022]

QATestLab, 2018. *Scenario-Based Testing: a brief overview.* [online] Tersedia di: <[https://qatestlab.com/resources/knowledge-center/scenario-based-](https://qatestlab.com/resources/knowledge-center/scenario-based-testing/)*[testing](https://qatestlab.com/resources/knowledge-center/scenario-based-testing/)*[/](https://qatestlab.com/resources/knowledge-center/scenario-based-testing/)> [Diakses 10 Agustus 2022]

Sommerville, I., 2021. *An Introduction to Modern Software Engineering, Global Edition.* London: Pearson Education.

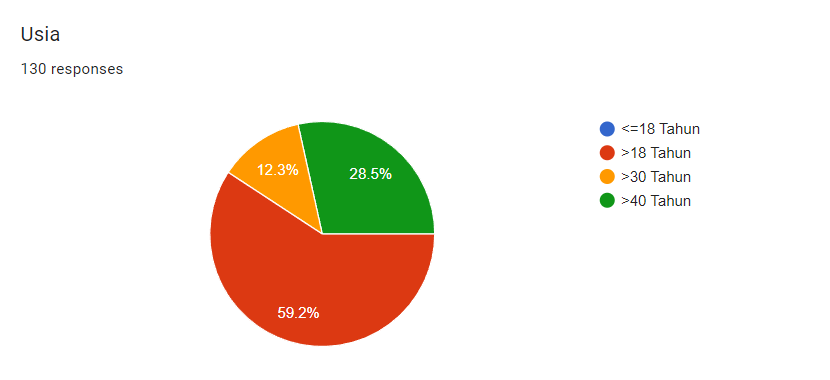
Ependi, U., Kurniawan, T. B., & Panjaitan, F., 2019. System *Usability* Scale Vs Heuristic Evaluation: A Review. *Jurnal Teknik Industri, Mesin, Elektro dan Ilmu Komputer,* [e-journal]10(1), 65-74. Tersedia melalui: Perreferensian Universitas Muria Kudus <<https://jurnal.umk.ac.id/index.php/simet/article/view/2725/1654>> [Diakses 18 Agustus 2022]

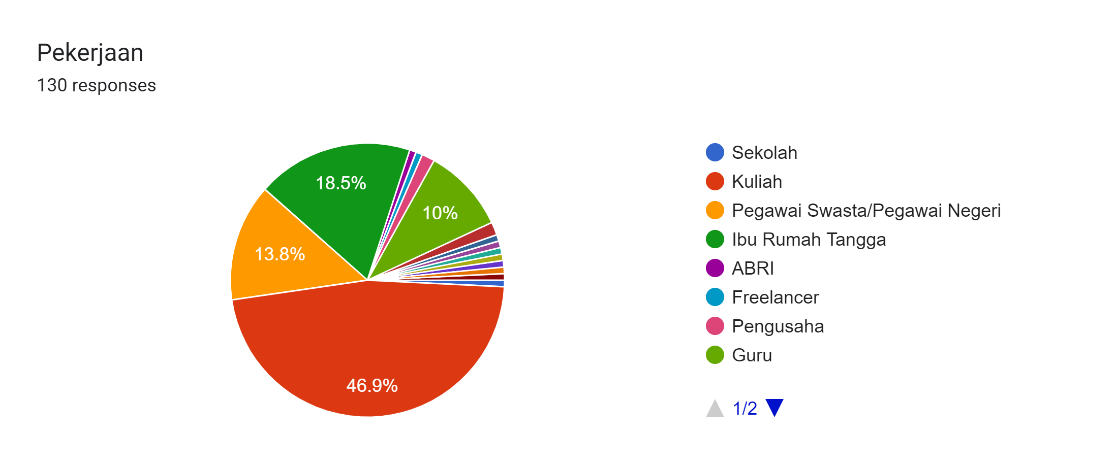
Brooke, J., 1995. *SUS: A Quick and Dirty Usability Scale.* [e-book] ResearchGate. Tersedia melalui: Website ResearchGate <[https://www.researchgate.net/publication/228593520\_SUS\_A\_quick\_and\_dirty\_](https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale)*[usability](https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale)*[\_scale](https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale)>[Diakses 18 Agustus 2022]

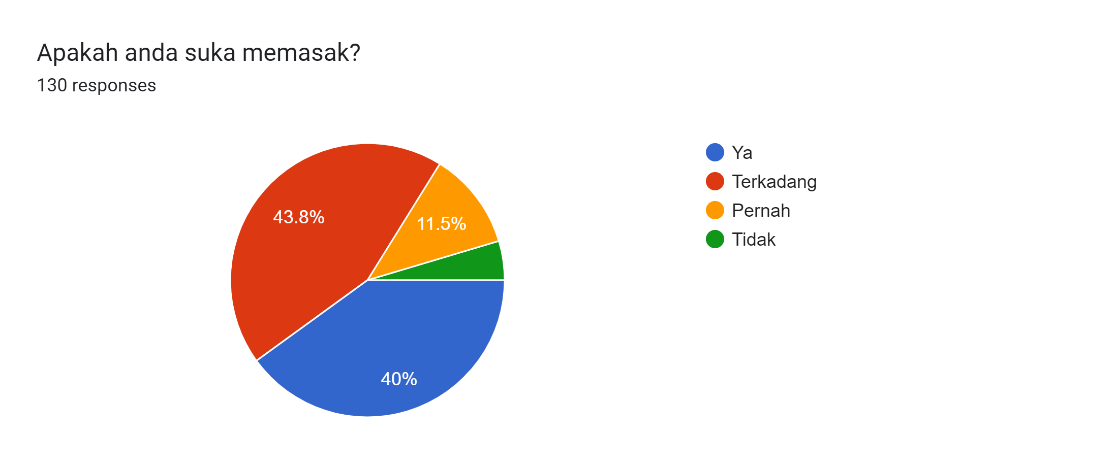
Romanpichler, 2014. *From Personas to User Stories.* [online] Tersedia di: <<https://www.romanpichler.com/blog/personas-epics-user-stories/>> [Diakses 21 Agustus 2022]

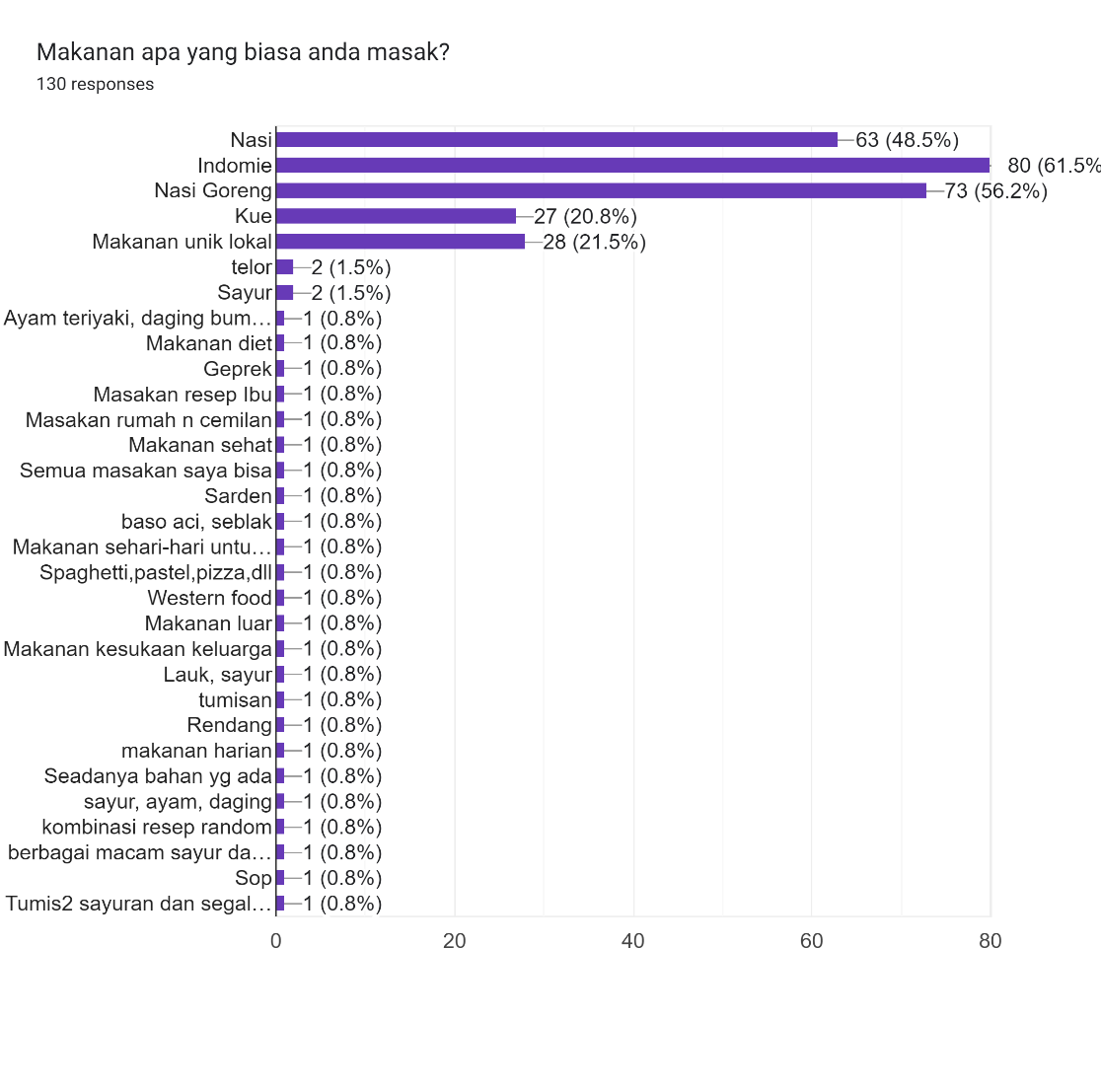
SoapUI, 2022. *Best Practices: Scenario-Based Testing.* [online] Tersedia di: <[https://www.soapui.org/learn/functional-](https://www.soapui.org/learn/functional-testing/scenario-based-testing/)*[testing](https://www.soapui.org/learn/functional-testing/scenario-based-testing/)*[/scenario-based-](https://www.soapui.org/learn/functional-testing/scenario-based-testing/)*[testing](https://www.soapui.org/learn/functional-testing/scenario-based-testing/)*[/](https://www.soapui.org/learn/functional-testing/scenario-based-testing/)> [Diakses 21 Agustus 2022]

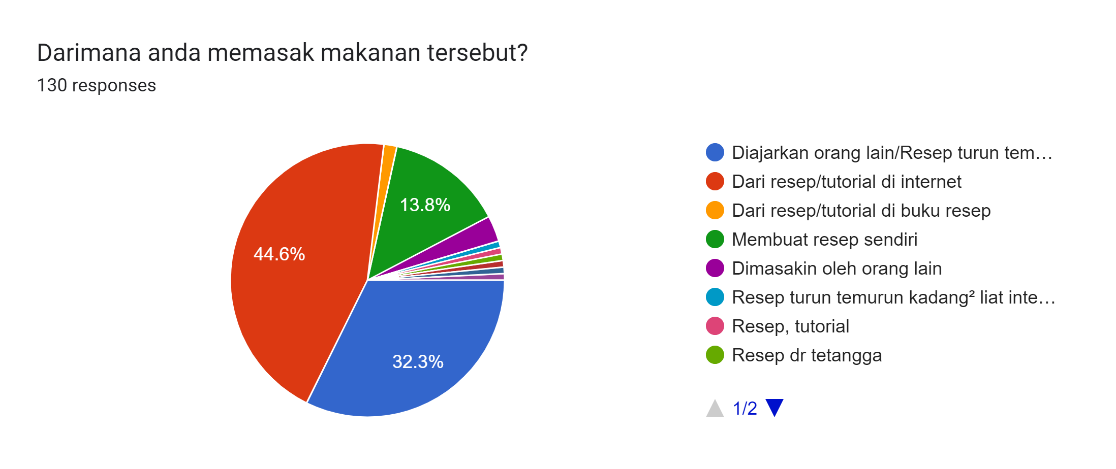
1. KUESIONER
   1. Kuesioner Masyarakat Umum Tentang Resep Masakan

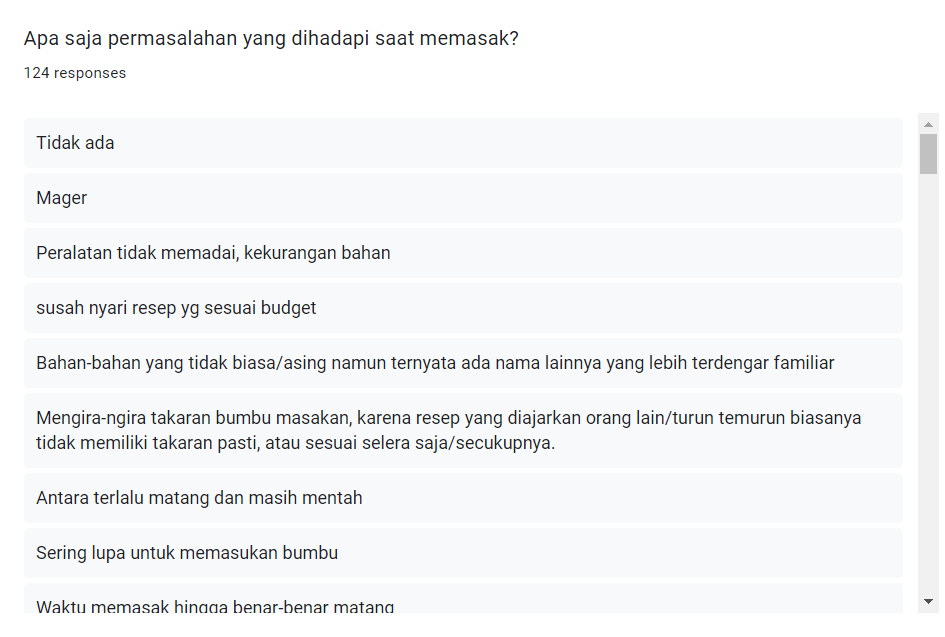




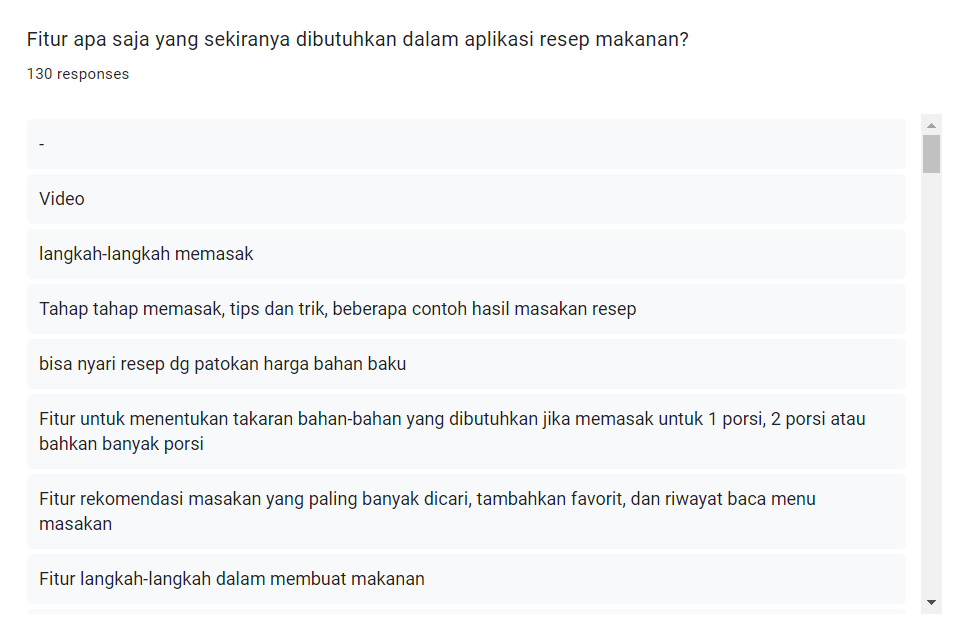




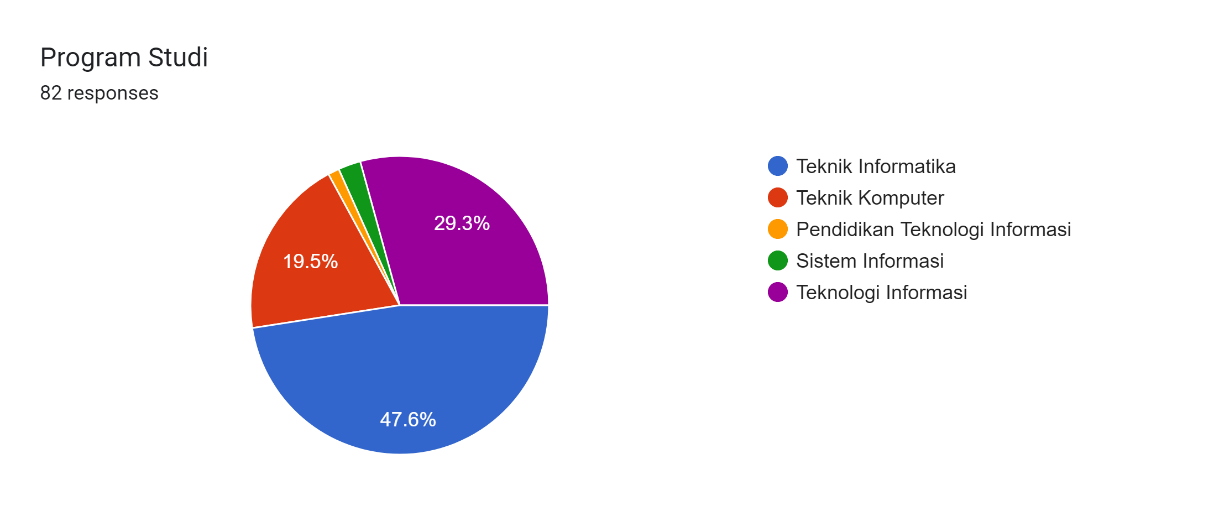


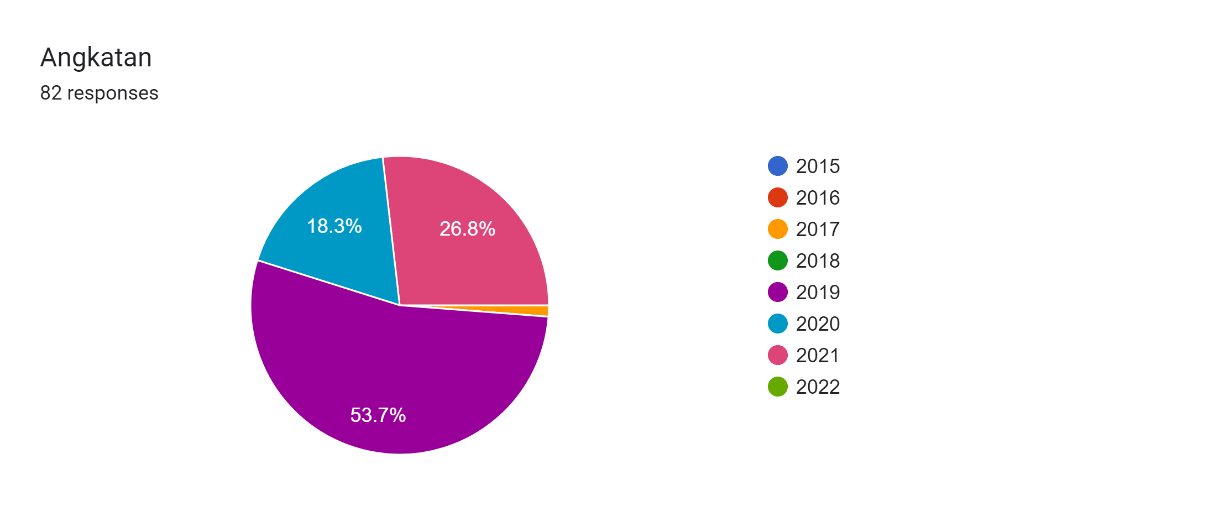


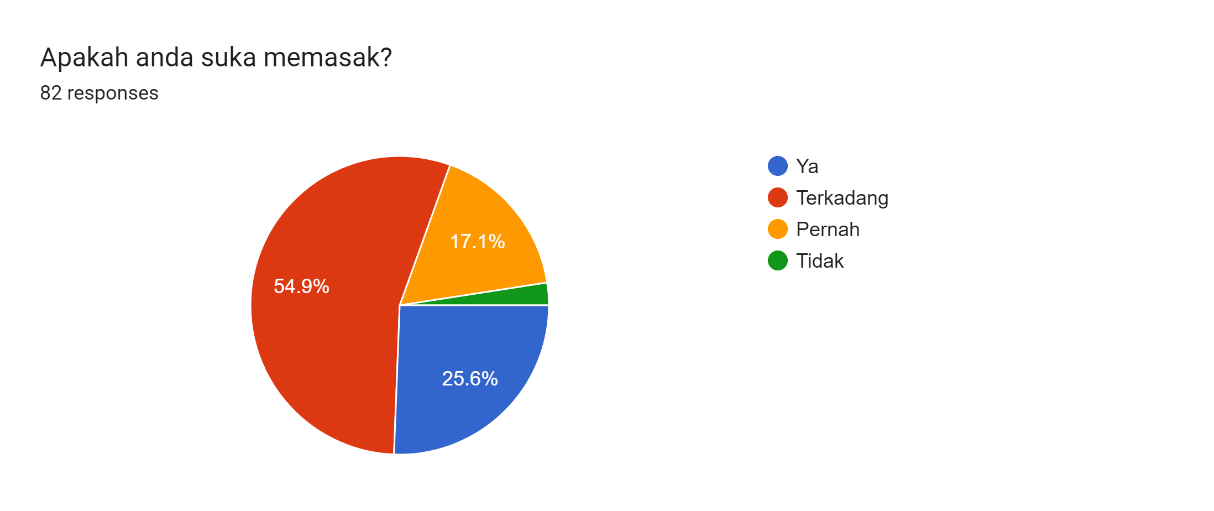


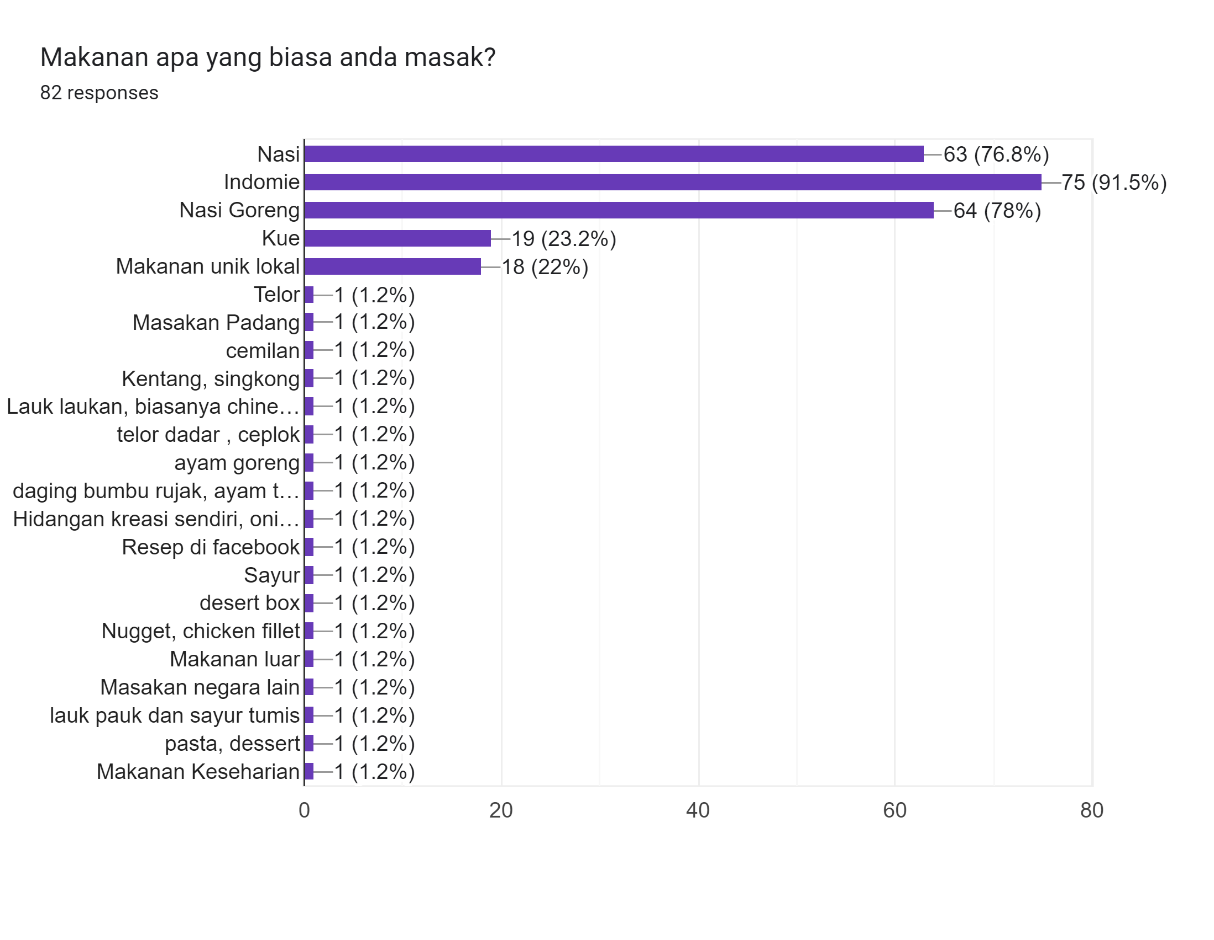


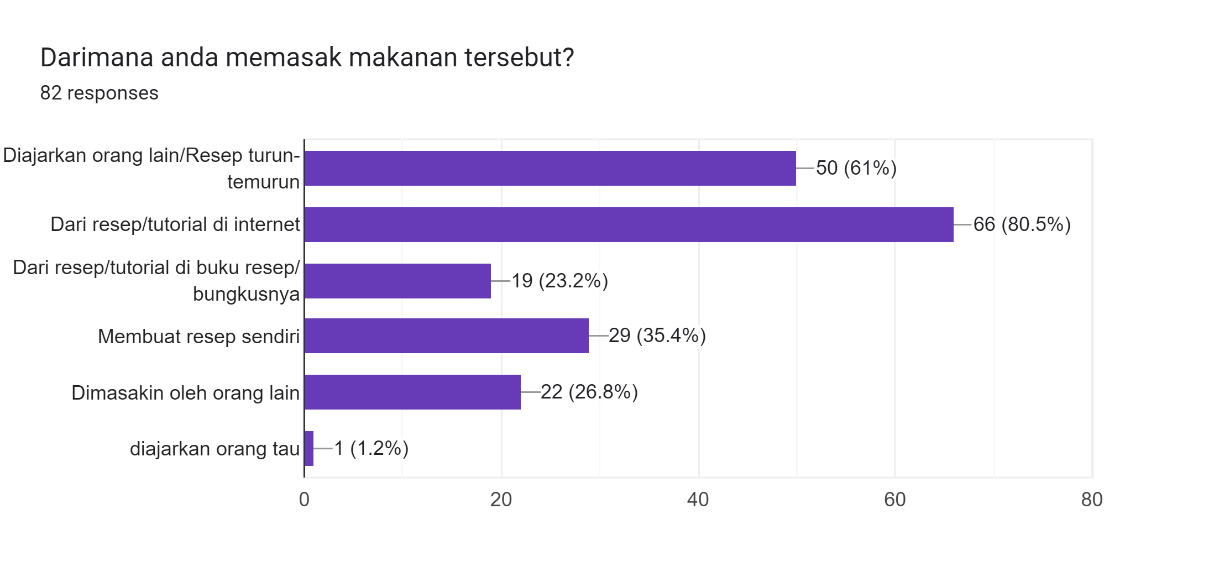
* 1. Kuesioner Mahasiswa Filkom Universitas Brawijaya Tentang Resep Masakan

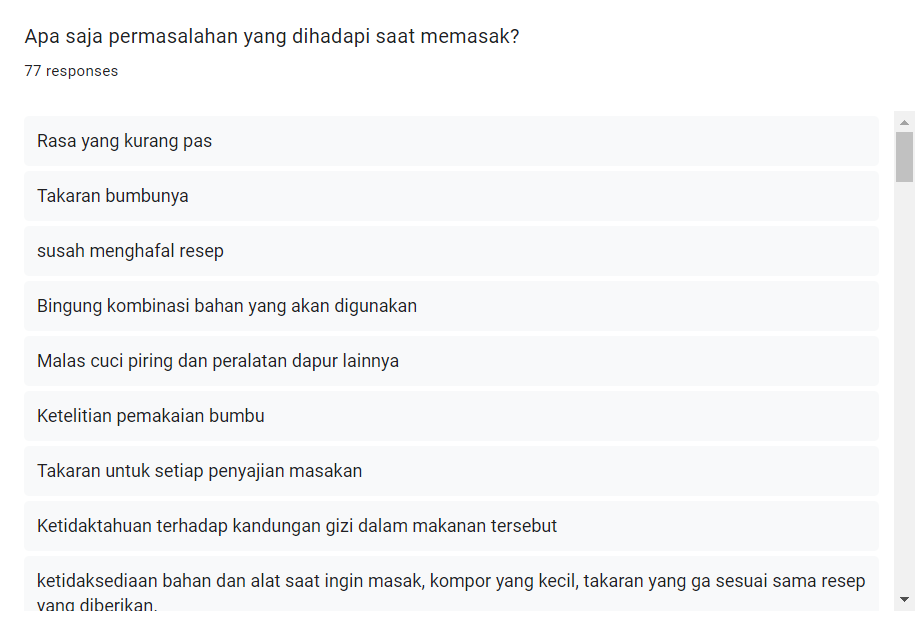




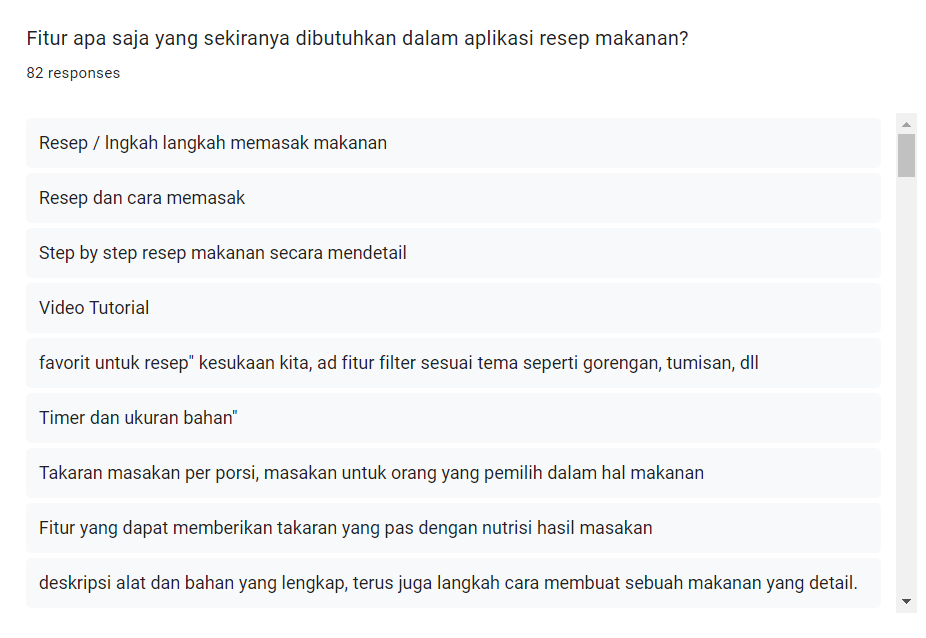












1. WAWANCARA
   1. Wawancara Analisis Kebutuhan Tentang Resep Masakan
2. Narasumber 1

- Nama : Munirotul Millah

- Tahun : 2022

- Judul : Wawancara Analisis Kebutuhan Tentang Resep Masakan

- Diwawancarai oleh : Muhammad Ziyad Al Maududi

- Teknis wawancara : Tatap muka secara langsung

- Tanggal wawancara : 19 September 2022

- Link rekaman wawancara :

<https://drive.google.com/file/d/1VtnI6xfutVLUaPN7rHyKvtDcKG5y6IVE/view?usp=sharing>

|  |  |
| --- | --- |
| **Pertanyaan** | **Jawaban** |
| Nama | Munirotul Millah. |
| Program Studi | Teknik Informatika. |
| Angkatan | 2019. |
| Apakah anda suka memasak? | Iya, akhir-akhir ini suka, sejak kuliah dan ngekos. |
| Makanan apa yang biasa anda masak? | Mie Goreng, Nasi Goreng, Kue Basah, Ayam Teriyaki, dan Daging Bumbu Rujak. |
| Darimana anda belajar memasak makanan tersebut? | Dari resep orang tua, dan *browsing* sendiri di google. |
| Apa saja permasalahan yang dihadapi saat memasak? | Tulisan bumbunya secukupnya, susah untuk takaran bumbu yang benar, resep dari orang tua tidak punya takaran asli atau selera sendiri. |
| Menurut anda, bagaimana cara anda mengatasi permasalahan tersebut? | Manual, tanya ke orang tua, atau kalau gagal gaenak bikin lagi, belajar dari kesalahan. |
| Apakah menu masakan yang terlalu rumit dapat mempersulit dalam memasak? | Iyasih, karena menu yang rumit bumbunya juga rumit, cara masaknya juga rumit. |
| Apakah jika ada aplikasi yang memberikan resep masakan dan panduan memasaknya, anda akan terbantu dalam mengatasi permasalahan memasak yang mungkin terjadi? | Menurut saya membantu sih, soalnya kalau cari di internet gak yang langsung dapat, diklik, pakai resep itu dilihat-lihat dulu, kalau sudah ada aplikasinya, dimana resepnya recommended dan bisa dipercaya, bisa lebih mempercepat masak kita sendiri. |
| Jika resep yang disajikan dalam aplikasi ini sederhana dan tidak terlalu rumit, yang mana lebih spesifik ke mahasiswa, apakah lebih terbantu? | Oh iya, terbantu sekali, apalagi kalau kuliah jauh ngerantau, biasa masak sendirian. |
| Apa saja yang akan anda butuhkan dan ingin lakukan dari aplikasi resep tersebut? | Rekomendasi resep makanan, fitur rating, populer, dan review terhadap resep yang pernah dimasak. |
| Apakah ada kritik dan saran? | Tidak usah terlalu rumit untuk fiturnya, yang penting tampilannya bagus, jadi biasanya untuk anak-anak muda prefer milih yang tampilannya bagus. |

1. Narasumber 2

- Nama : Gilbert Samuel Nicholas Silaban

- Tahun : 2022

- Judul : Wawancara Analisis Kebutuhan Tentang Resep Masakan

- Diwawancarai oleh : Muhammad Ziyad Al Maududi

- Teknis wawancara : Tatap muka secara langsung

- Tanggal wawancara : 19 September 2022

- Link rekaman wawancara :

<https://drive.google.com/file/d/1VxXJnsq9r9WDrWD1VpFMKHtI0A_NEmlf/view?usp=sharing>

|  |  |
| --- | --- |
| **Pertanyaan** | **Jawaban** |
| Nama | Gilbert Samuel Nicholas Silaban. |
| Program Studi | Teknik Informatika. |
| Angkatan | 2019. |
| Apakah anda suka memasak? | Kalau saya suka memasak juga. |
| Makanan apa yang biasa anda masak? | Indomie, spageti, atau telur. |
| Darimana anda belajar memasak makanan tersebut? | Saya biasanya belajar memasak itu dari melihat tutorial di youtube, atau melihat cara penyajian di belakang bungkus kemasan. |
| Apa saja permasalahan yang dihadapi saat memasak? | Ketika melihat suatu resep ada kayak istilah-istilah asing dimana kita harus melihat dulu di Google, atau melihat dulu videonya, lebih ke istilah di resep yang kurang jelas, seperti mouisture gitu. |
| Menurut anda, bagaimana cara anda mengatasi permasalahan tersebut? | Video tutorial memasak, dapat dilihat takaran yang sesuai dan ukurannya berapa, seperti misalnya secukupnya atau sejumput. |
| Apakah menu masakan yang terlalu rumit dapat mempersulit dalam memasak? | Sangat mempersulit. |
| Apakah jika ada aplikasi yang memberikan resep masakan dan panduan memasaknya, anda akan terbantu dalam mengatasi permasalahan memasak yang mungkin terjadi? | Sangat terbantu. |
| Jika resep yang disajikan dalam aplikasi ini sederhana dan tidak terlalu rumit, yang mana lebih spesifik ke mahasiswa, apakah lebih terbantu? | Bisa jadi. |
| Apa saja yang akan anda butuhkan dan ingin lakukan dari aplikasi resep tersebut? | Aplikasi resep tersebut bisa kita lihat dulu daftar menunya, apa saja bahannya, berapa takarannya, video tutorial memasaknya, perkiraan berapa porsi, dan berapa harga yang dikeluarkan. |
| Apakah ada kritik dan saran? | Untuk ke depannya mungkin bisa dibuat secara iteratif, dan seperti *live cooking* juga boleh. |

1. Narasumber 3

- Nama : Anargya Hassya

- Tahun : 2022

- Judul : Wawancara Analisis Kebutuhan Tentang Resep Masakan

- Diwawancarai oleh : Muhammad Ziyad Al Maududi

- Teknis wawancara : Tatap muka secara langsung

- Tanggal wawancara : 19 September 2022

- Link rekaman wawancara :

<https://drive.google.com/file/d/1VxXJnsq9r9WDrWD1VpFMKHtI0A_NEmlf/view?usp=sharing> <https://drive.google.com/file/d/1L16jit4t4ve8gyt9LV1wVoYw95DcI_WB/view?usp=sharing>

|  |  |
| --- | --- |
| **Pertanyaan** | **Jawaban** |
| Nama | Anargya Hassya. |
| Program Studi | Teknik Informatika. |
| Angkatan | 2019. |
| Apakah anda suka memasak? | Lumayan suka. |
| Makanan apa yang biasa anda masak? | Nasi Goreng, Bakso, dan Indomie. |
| Darimana anda belajar memasak makanan tersebut? | Saya biasa lihat di Google, atau tutorial Youtube. |
| Apa saja permasalahan yang dihadapi saat memasak? | Mungkin terkadang lupa step-stepnya, barang-barang belanjaan juga tidak ada di pasar. |
| Menurut anda, bagaimana cara anda mengatasi permasalahan tersebut? | Untuk step-stepnya searching lagi di youtube, misal ada yang kurang nanti saya cari-cari lagi, untuk yang bahan itu harus keliling-keliling pasar lagi. |
| Apakah menu masakan yang terlalu rumit dapat mempersulit dalam memasak? | Iya, karena setiap orang punya kemampuan masak yang berbeda, jadi tingkat kerumitan suatu resep itu pasti punya pengaruh terhadap pembuatan makanan itu. |
| Apakah jika ada aplikasi yang memberikan resep masakan dan panduan memasaknya, anda akan terbantu dalam mengatasi permasalahan memasak yang mungkin terjadi? | Jelas ya, untuk resep jadi saya tahu bahan-bahan yang saya pakai, dan panduan-panduan memasaknya lengkap, juga saya akan mudah dalam memasak makanan sehari-hari. |
| Jika resep yang disajikan dalam aplikasi ini sederhana dan tidak terlalu rumit, yang mana lebih spesifik ke mahasiswa, apakah lebih terbantu? | Oh iya, untuk mahasiswa mungkin bisa dibuat lebih sederhana saja. |
| Apa saja yang akan anda butuhkan dan ingin lakukan dari aplikasi resep tersebut? | Mungkin sejauh ini saya hanya butuh panduan memasak, dan resepnya saja sih. |
| Apakah ada kritik dan saran? | Mungkin saran, untuk UI aplikasi dibuat lebih sederhana saja, yang penting on-point saja penyampaian resepnya. |