

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

PROPOSAL SKRIPSI

Disusun oleh:
Muhammad Ziyad Al Maududi
NIM: 195150200111067



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2022

DAFTAR ISI

DAFTAR ISI	ii
DAFTAR TABEL.....	iv
DAFTAR GAMBAR.....	v
DAFTAR LAMPIRAN	vi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	4
1.6 Sistematika Pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Memasak.....	5
2.2.1 Resep Masakan	5
2.3 <i>Prototyping</i>	6
2.4 <i>Personas</i>	7
2.5 <i>Scenarios</i>	9
2.6 <i>User Stories</i>	10
2.7 <i>Features</i>	11
2.8 Aplikasi Perangkat Bergerak	16
2.9 Android	16
2.10 Android Studio	17
2.11 Firebase.....	17
2.12 Arsitektur Pengembangan Aplikasi Android	19
2.13 Pengujian Perangkat Lunak	21
2.13.1 <i>Scenario-Based Testing</i>	21
2.13.2 <i>Usability Testing</i>	23
BAB 3 METODOLOGI	26

3.1 Tipe Penelitian	26
3.2 Metode Penelitian	26
3.3 Subjek Penelitian	28
3.4 Lokasi Penelitian	28
3.5 Teknik Pengumpulan Data	28
3.6 Peralatan Pendukung yang Digunakan	28
3.7 Metode Pengembangan	28
3.7.1 Studi Literatur, Survey, Wawancara	29
3.7.2 Analisis Kebutuhan	29
3.7.3 Pengembangan <i>Prototype</i>	30
3.7.4 Evaluasi <i>Prototype</i>	30
3.7.5 Perancangan	31
3.7.6 Implementasi	31
3.7.7 Pengujian	31
3.7.8 Kesimpulan dan Saran	32
DAFTAR REFERENSI	33
LAMPIRAN A KUESIONER MASYARAKAT UMUM	38
A.1 Kuesioner Masyarakat Umum Tentang Resep Masakan	38
A.2 Kuesioner Masyarakat Umum Tentang Resep Masakan	41

DAFTAR TABEL

Tabel 2.1 Dependensi Firebase Authentication	18
Tabel 2.2 Dependensi FirebaseUI	18
Tabel 2.3 Dependensi Firebase Cloud Firestore	18
Tabel 2.4 Dependensi Firebase Cloud Storage	19
Tabel 2.5 Instrumen pengujian <i>System Usability Scale</i>	23
Tabel 3.1 Daftar Pertanyaan Wawancara untuk Analisis Kebutuhan	29
Tabel 3.2 Daftar Pertanyaan Wawancara untuk Evaluasi <i>Prototype</i>	30
Tabel 3.3 Daftar Pertanyaan Kuesioner untuk Pengujian	31

DAFTAR GAMBAR

Gambar 2.1 Tahapan dari <i>Personas</i> hingga menjadi <i>Features</i>	7
Gambar 2.2 Deskripsi sebuah <i>Personas</i>	8
Gambar 2.3 Contoh <i>Personas</i> dalam bentuk deskripsi naratif	9
Gambar 2.6 Deskripsi sebuah <i>Scenarios</i>	9
Gambar 2.8 Contoh <i>Scenarios</i>	10
Gambar 2.9 Format standar kalimat <i>User Stories</i>	11
Gambar 2.10 Contoh kalimat <i>User Stories</i> dengan format standar	11
Gambar 2.11 Format varian standar kalimat <i>User Stories</i> dengan justifikasi	11
Gambar 2.12 Contoh kalimat <i>User Stories</i> dengan format varian standar	11
Gambar 2.13 Contoh <i>User Stories</i> dari sebuah <i>Scenarios</i>	11
Gambar 2.14 Deskripsi sebuah <i>Features</i>	12
Gambar 2.15 Format kalimat deskripsi sebuah <i>Features</i>	12
Gambar 2.16 Desain <i>Features</i>	13
Gambar 2.17 Faktor dalam set desain <i>Features</i>	14
Gambar 2.18 Pertanyaan terhadap <i>Features</i> untuk menghindari <i>feature creep</i> ..	14
Gambar 2.19 Melakukan <i>highlight</i> frasa terhadap <i>Scenarios</i>	15
Gambar 2.20 Contoh penentuan <i>Features</i> berdasarkan <i>Scenarios</i>	15
Gambar 2.21 Contoh <i>Features</i> menggunakan template model <i>input/action/output</i>	16
Gambar 2.23 Komponen aplikasi Android	17
Gambar 2.24 Rancangan arsitektur <i>MVVM</i>	21
Gambar 2.25 Contoh skenario pada <i>scenario-based testing</i>	22
Gambar 2.26 Contoh sebuah <i>scenario-based testing</i>	22
Gambar 2.27 Skala penilaian <i>System Usability Scale</i>	24
Gambar 2.28 Langkah-langkah <i>System Usability Scale</i>	24
Gambar 2.29 Penilaian <i>System Usability Scale</i>	25
Gambar 3.1 Metode penelitian	27

DAFTAR LAMPIRAN

LAMPIRAN A KUESIONER MASYARAKAT UMUM.....	38
A.1 Kuesioner Masyarakat Umum Tentang Resep Masakan.....	38
A.2 Kuesioner Masyarakat Umum Tentang Resep Masakan.....	41

BAB 1 PENDAHULUAN

1.1 Latar Belakang

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

Berbagai media yang dapat digunakan dalam mencari resep masakan antara lain membaca buku mengenai resep masakan, menonton acara memasak di televisi, maupun mengikuti kursus memasak. Namun, sebagian orang menganggap cara-cara tersebut sangat merepotkan, sehingga mereka menggunakan *smartphone* untuk mengakses resep makanan tersebut via aplikasi (Karlina, Asian, & Mahmud, 2019). Selain itu, resep masakan yang hanya didapatkan secara turun-temurun terkadang menjadi terlupakan akibat padatnya aktivitas seseorang. Oleh karena itu, dibutuhkan resep masakan yang dapat diakses kapan pun saat dibutuhkan (Santoso, 2016). Di era globalisasi dengan semakin canggihnya kemampuan teknologi, berbagai resep masakan diposting setiap waktu oleh beberapa orang dari penjuru Indonesia melalui internet. Bahkan, kita dapat mengaksesnya tanpa harus menggunakan komputer, yaitu dengan menggunakan *smartphone/handphone*. Salah satu *platform smartphone* terpopuler saat ini yaitu Android. Beberapa aplikasi Android 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. Alur Metodologi Penelitian dalam *SDLC Prototyping* terdiri dari Studi Literatur, Survey, Wawancara, Analisis Kebutuhan, Pengembangan *Prototype*, Evaluasi *Prototype*, Perancangan, Implementasi, dan Pengujian.

Alasan penggunaan aplikasi perangkat bergerak sebagai solusi dari permasalahan tersebut yaitu karena agar aplikasi dapat diakses dimana saja. 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).

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

1.2 Rumusan Masalah

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

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

1.3 Tujuan

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

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

1.4 Manfaat

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

1. Untuk Peneliti:
 - a. Mengaplikasikan ilmu yang didapat selama perkuliahan
 - b. Menambah pengetahuan, keterampilan, dan pengalaman dalam mengembangkan aplikasi perangkat bergerak berbasis Android
 - c. Penelitian ini dapat menjadi sumber literatur penelitian selanjutnya
2. Untuk Pengguna:
 - a. Memberikan resep dan panduan dalam memasak, dalam hal ini masakan sederhana yang sesuai dengan mahasiswa

1.5 Batasan Masalah

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

1. Aplikasi ini dibuat untuk *platform* sistem operasi Android dengan *API* 23 sampai 32.
2. Penggunaan aplikasi memerlukan koneksi internet.
3. Aplikasi ini dibuat menggunakan bahasa pemrograman Kotlin, berbasis Android, dikembangkan dan dijalankan menggunakan Android Studio.
4. Mahasiswa yang dimaksud dalam penelitian ini memiliki studi kasus dalam lingkup mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya.

1.6 Sistematika Pembahasan

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

1. BAB 1 – PENDAHULUAN
Bab ini digunakan untuk menjelaskan latar belakang penelitian, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.
2. BAB 2 - LANDASAN KEPUSTAKAAN
Bab ini berisi tentang kajian pustaka atau dasar teori yang digunakan dalam penyelesaian permasalahan di penelitian ini, teknologi pendukung, dan uraian hasil penelitian-penelitian terdahulu yang berhubungan dengan penelitian ini.
3. BAB 3 – METODOLOGI PENELITIAN
Bab ini berisi tentang cara sistematis dalam penyelesaian masalah, yang mana mencakup tipe penelitian, metode penelitian, subjek penelitian, lokasi penelitian, teknik pengumpulan data, peralatan pendukung yang digunakan, dan metode pengembangan.
4. BAB 4 - ANALISIS KEBUTUHAN
Bab ini berisi tentang deskripsi sistem, dan analisis kebutuhan perangkat lunak.
5. BAB 5 – PERANCANGAN
Bab ini berisi tentang perancangan sistem yang mengacu pada hasil analisis kebutuhan perangkat lunak.
6. BAB 6 – IMPLEMENTASI
Bab ini berisi tentang implementasi dari perancangan perangkat lunak.
7. BAB 7 – PENGUJIAN
Bab ini berisi tentang pengujian dari perangkat lunak.
8. BAB 8 – PENUTUP
Bab ini berisi tentang kesimpulan dan saran.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Dalam melakukan penelitian, langkah awal penyelesaian permasalahan yaitu dengan mencari tahu langkah-langkah yang diperlukan yang bersumber dari penelitian-penelitian terdahulu yang terkait dengan penelitian yang dilakukan (Muchlisson, Kharisma, & Arwani, 2022).

Terdapat beberapa penelitian terdahulu yang berhubungan dengan aplikasi 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*, dengan bahasa pemrograman yang digunakan yaitu Java. Penelitian ini menghasilkan sebuah aplikasi yang membantu dalam memasak, dan cukup efisien dalam penggunaannya, dengan tampilan aplikasi yang cukup baik. Perbedaannya dengan penelitian ini yaitu pengembangan perangkat lunak berfokus pada masakan khas sebuah daerah, sedangkan pada penelitian ini difokuskan pada masakan sederhana yang cocok dengan mahasiswa.

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*, 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*.

2.2 Memasak

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).

2.2.1 Resep Masakan

Resep masakan adalah sebuah takaran yang digunakan untuk mengolah bahan makanan yang keakuratannya telah teruji. Takaran yang dimaksud dapat berupa bumbu, bahan, maupun cara pengolahan bahan makanan (Arifien, 2022). Berbagai media yang dapat digunakan dalam mencari resep masakan antara lain membaca buku mengenai resep masakan, menonton acara memasak di televisi,

maupun mengikuti kursus memasak. Namun, sebagian orang menganggap cara-cara tersebut sangat merepotkan, sehingga mereka menggunakan *smartphone* untuk mengakses resep makanan tersebut via aplikasi (Karlina, Asian, & Mahmud, 2019). Selain itu, resep masakan yang hanya didapatkan secara turun-temurun terkadang menjadi terlupakan akibat padatnya aktivitas seseorang. Oleh karena itu, dibutuhkan resep masakan yang dapat diakses kapan pun saat dibutuhkan (Santoso, 2016).

2.3 Prototyping

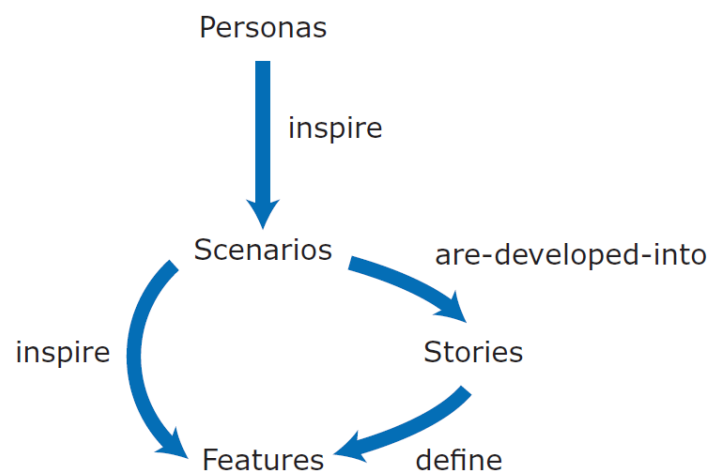
Pengembangan perangkat lunak adalah sebuah proses membuat sebuah perangkat lunak baru untuk menggantikan yang lama secara keseluruhan, atau memperbaiki perangkat lunak yang sudah ada. Dibutuhkan sebuah metodologi khusus agar solusi dan pengembangan perangkat lunak lebih cepat, tepat, dan mudah dikembangkan 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 terjadinya krisis perangkat lunak. Hal itu dikarenakan metode *SDLC* memiliki tahapan yang sangat detail, mulai dari analisis, hingga pemeliharaan. *SDLC* juga dapat memudahkan programmer mengembangkan sistem, mengurangi kesalahan, dan mempercepat pengerjaan (Aisah, et al., 2021).

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).



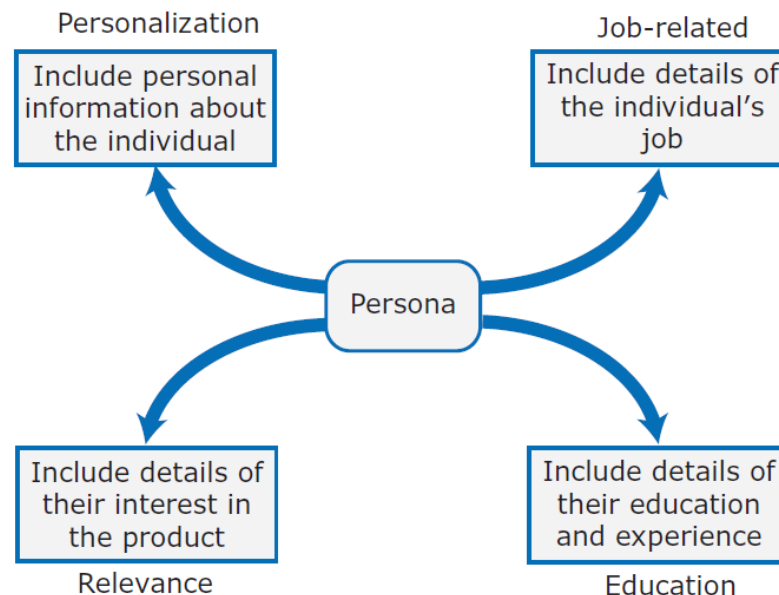
Gambar 2.1 Tahapan dari *Personas* hingga menjadi *Features*

Sumber: Sommerville (2021)

2.4 *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).

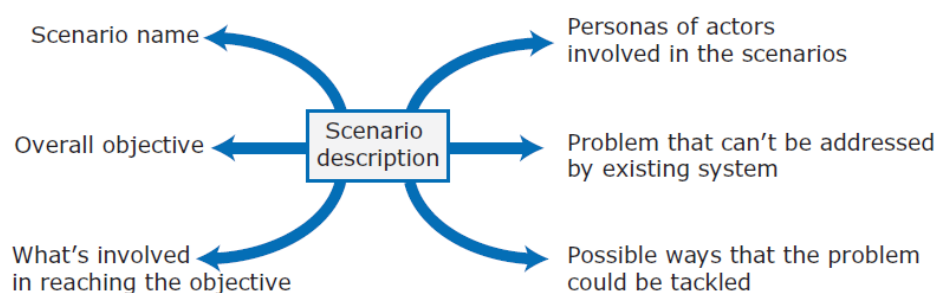
Emma, a history teacher
<p>Emma, age 41, is a history teacher in a secondary school (high school) in Edinburgh. She teaches students from ages 12 to 18. She was born in Cardiff in Wales, where both her father and her mother were teachers. After completing a degree in history from Newcastle University, she moved to Edinburgh to be with her partner and trained as a teacher. She has two children, aged 6 and 8, who both attend the local primary school. She likes to get home as early as she can to see her children, so often does lesson preparation, administration, and marking from home.</p> <p>Emma uses social media and the usual productivity applications to prepare her lessons, but is not particularly interested in digital technologies. She hates the virtual learning environment that is currently used in her school and avoids using it if she can. She believes that face-to-face teaching is most effective. She might use the iLearn system for administration and access to historical films and documents. However, she is not interested in a blended digital/face-to-face approach to teaching.</p>

Gambar 2.3 Contoh *Personas* dalam bentuk deskripsi naratif

Sumber: Sommerville (2021)

2.5 *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).

Fishing in Ullapool
Jack is a primary school teacher in Ullapool, teaching P6 pupils. He has decided that a class project should be focused around the fishing industry in the area, looking at the history, development, and economic impact of fishing.
As part of this, students are asked to gather and share reminiscences from relatives, use newspaper archives, and collect old photographs related to fishing and fishing communities in the area. Pupils use an iLearn wiki to gather together fishing stories and SCRAN (a history archive site) to access newspaper archives and photographs. However, Jack also needs a photo-sharing site as he wants students to take and comment on each others' photos and to upload scans of old photographs that they may have in their families. He needs to be able to moderate posts with photos before they are shared, because pre-teen children can't understand copyright and privacy issues.
Jack sends an email to a primary school teachers' group to see if anyone can recommend an appropriate system. Two teachers reply and both suggest that he use KidsTakePics, a photo-sharing site that allows teachers to check and moderate content. As KidsTakePics is not integrated with the iLearn authentication service, he sets up a teacher and a class account with KidsTakePics.
He uses the the iLearn setup service to add KidsTakePics to the services seen by the students in his class so that, when they log in, they can immediately use the system to upload photos from their phones and class computers.

Gambar 2.5 Contoh *Scenarios*

Sumber: Sommerville (2021)

2.6 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.

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

Gambar 2.6 Format standar kalimat *User Stories*

Sumber: Sommerville (2021)

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

Gambar 2.7 Contoh kalimat *User Stories* dengan format standar

Sumber: Sommerville (2021)

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

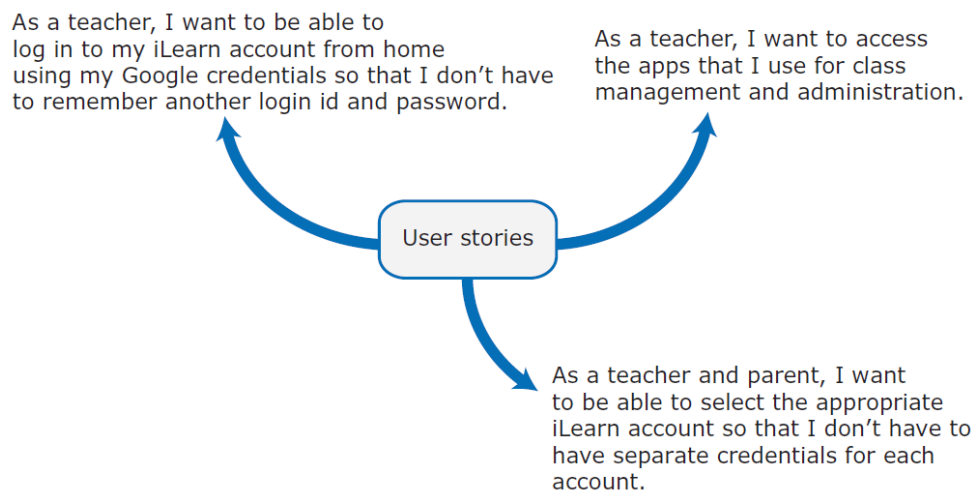
Gambar 2.8 Format varian standar kalimat *User Stories* dengan justifikasi

Sumber: Sommerville (2021)

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

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

Sumber: Sommerville (2021)



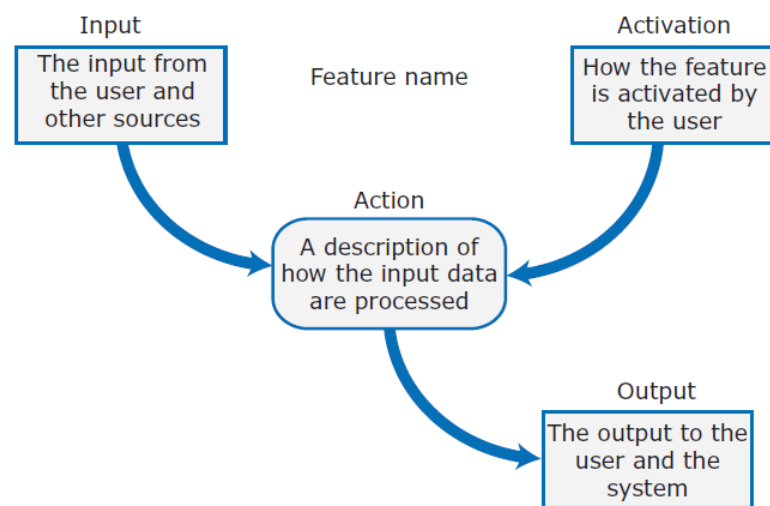
Gambar 2.10 Contoh *User Stories* dari sebuah *Scenarios*

Sumber: Sommerville (2021)

2.7 Features

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

fitur dari sebuah produk haruslah independen, koheren, dan relevan. Independen berarti sebuah fitur seharusnya tidak tergantung kepada bagaimana fitur sistem lain diimplementasikan, dan tidak terpengaruh oleh urutan aktivasi fitur lain. Koheren berarti fitur haruslah terhubung dengan satu item fungsionalitas, tidak boleh melakukan lebih dari satu hal, dan haruslah tidak memiliki efek samping. Relevan berarti fitur sistem haruslah mencerminkan cara yang biasanya pengguna melakukan beberapa tugas, dan tidak boleh menawarkan fungsionalitas tidak jelas yang jarang diperlukan (Sommerville, 2021).



Gambar 2.11 Deskripsi sebuah *Features*

Sumber: Sommerville (2021)

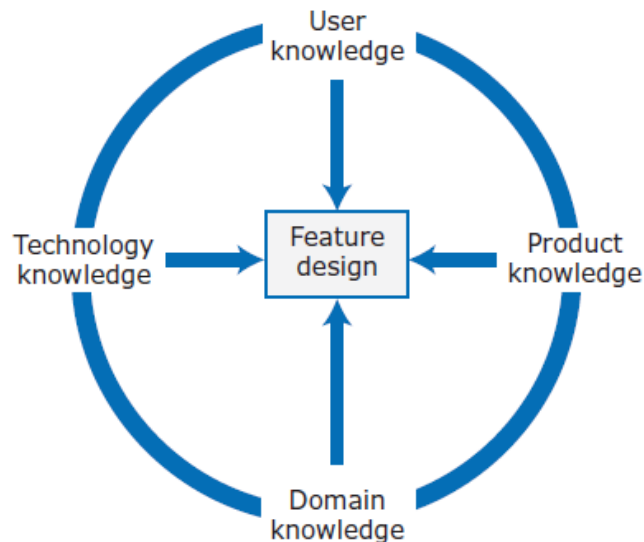
<action> the <result> <by/for/of/to> <object>

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

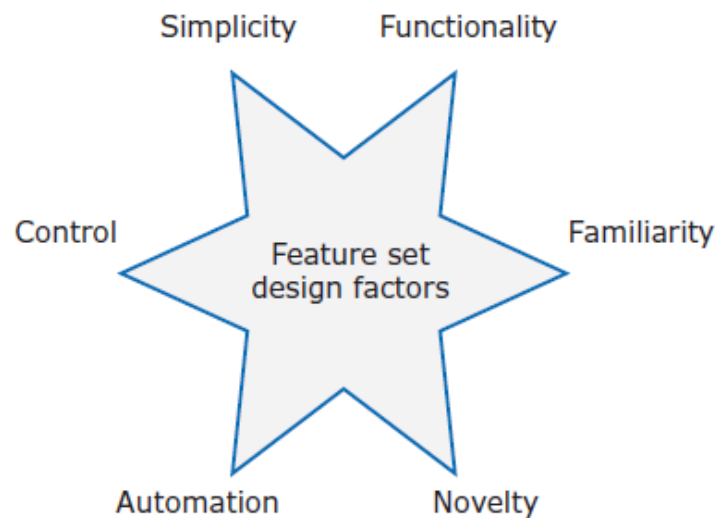
untuk mengambil keuntungan dari pengembangan teknologi sejak kompetitor hadir, dimana memanfaatkan teknologi terbaru dalam mendesain fitur yang menggunakannya. Namun, tidak semuanya begitu penting untuk semua jenis produk. Misalnya, pengetahuan domain sangat penting untuk produk bisnis, namun kurang penting untuk produk konsumen generik (Sommerville, 2021).



Gambar 2.13 Desain *Features*

Sumber: Sommerville (2021)

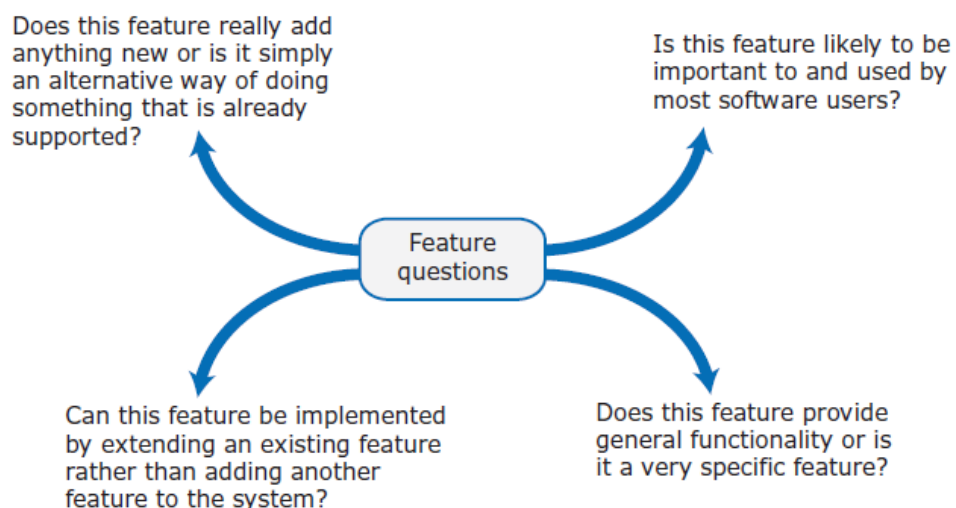
Ada 6 faktor yang perlu dipertimbangkan saat mendesain sebuah set fitur dari produk, dan menentukan bagaimana fitur-fitur tersebut bekerja. Namun, mustahil untuk menerapkan semua faktor tersebut secara optimal, sehingga diperlukan sejumlah pertukaran. Pertukaran-pertukaran dalam faktor tersebut antara 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 menggeneralisir 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).

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

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

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

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

Gambar 2.16 Melakukan *highlight* frasa terhadap *Scenarios*

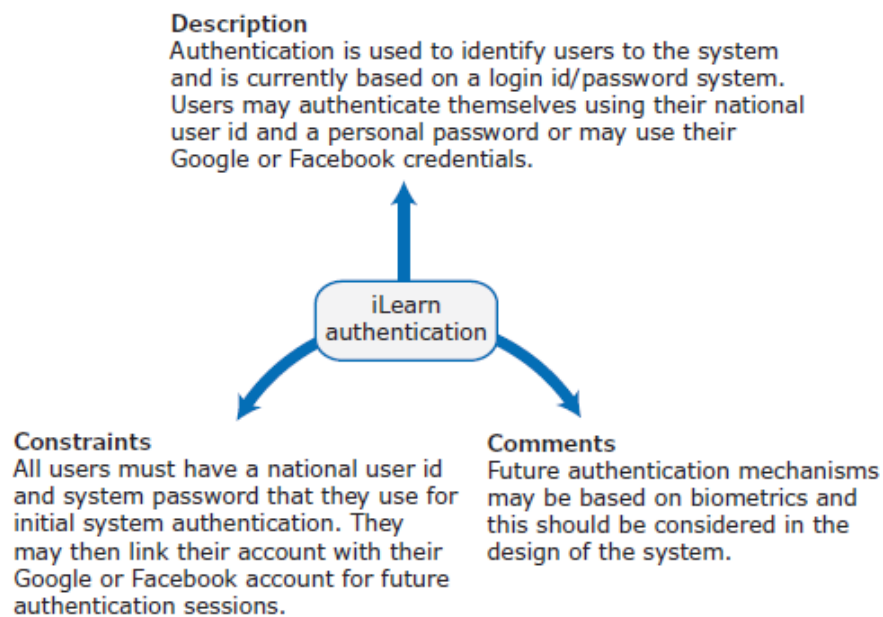
Sumber: Sommerville (2021)

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

Gambar 2.17 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)

2.8 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).

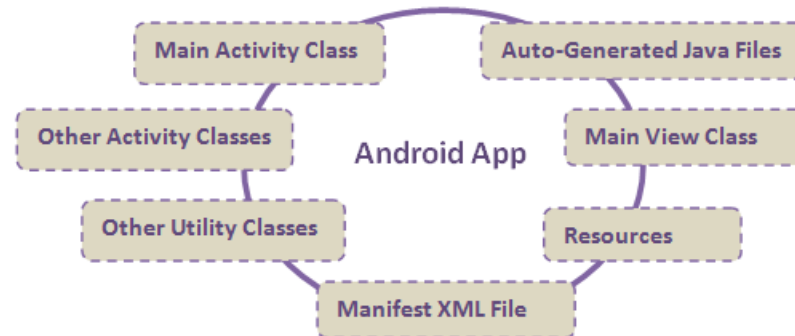
Salah satu bentuk aplikasi perangkat bergerak adalah aplikasi Android. Pemilihan Android sebagai bentuk aplikasi perangkat bergerak dalam penelitian ini dimaksudkan untuk menjangkau pengguna yang lebih banyak, dan didasari statistik mengenai jumlah pengguna sistem operasi Android di Indonesia yang mencapai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

2.9 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)

2.10 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).

2.11 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).

2.11.1.1 Firebase Authentication

Firebase Authentication adalah sebuah layanan sistem autentikasi yang aman, sekaligus meningkatkan pengalaman autentikasi dan orientasi pengguna. Layanan ini mendukung autentikasi menggunakan email dan 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)

2.11.1.2 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)

2.11.1.3 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)

2.11.1.4 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)

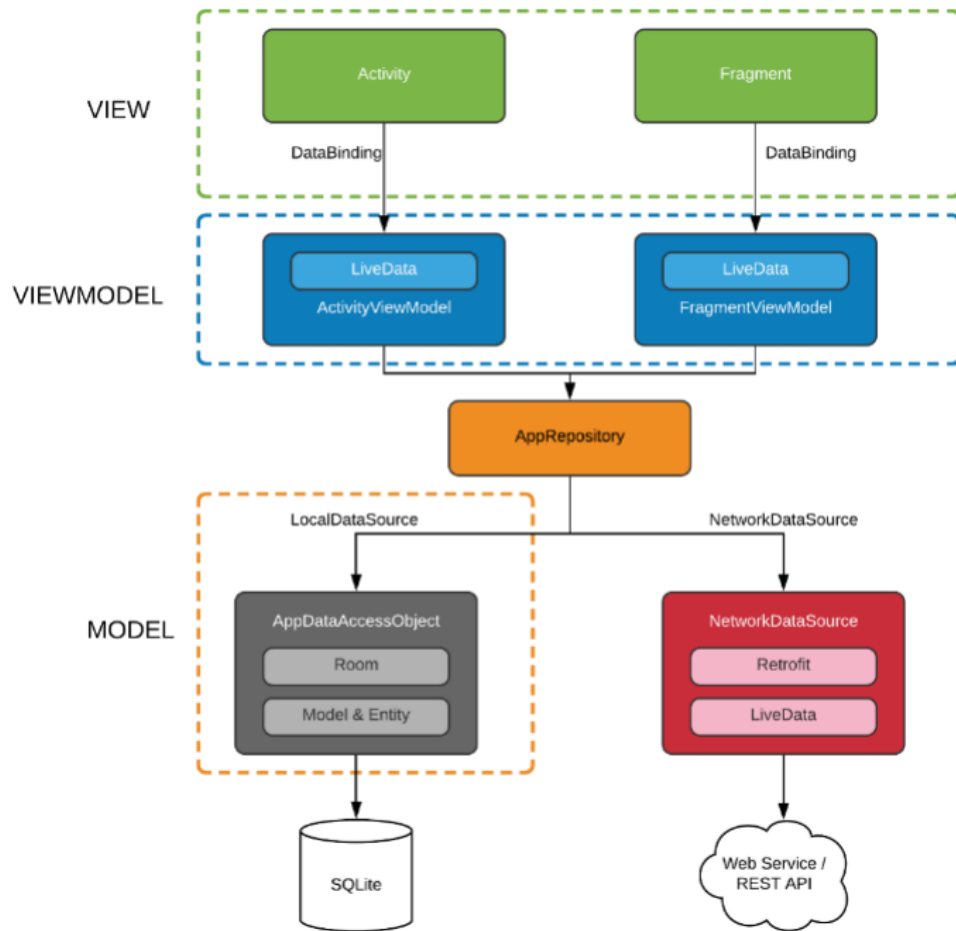
2.12 Arsitektur Pengembangan Aplikasi Android

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

Model-View-ViewModel atau *MVVM* adalah salah satu pola arsitektur yang mengusung pemisahan presentasi, yang mana memisahkan antara grafis antarmuka dan proses logika bisnis (Arif, Musthafa, & Muriyatmoko, 2019). Beberapa komponen dari pola *MVVM* yaitu *Model*, *View*, dan *ViewModel*. *Model* mewakili proses logika bisnis dan data, yang mana dalam hal ini proses bisnis logika menentukan bagaimana data dimanipulasi. *View* mewakili komponen grafis antarmuka, namun hanya bertanggung jawab mewakili data, tanpa memanipulasinya. Pada *MVVM*, *View* merupakan komponen aktif yang berisi perilaku, peristiwa, dan pengikatan data, yang mana membutuhkan informasi mengenai komponen *ViewModel*, dan *Model* yang mendasarinya. *ViewModel* adalah sebuah komponen yang membantu pemisahan presentasi, dalam hal ini

memisahkan *View* dan *Model*, sekaligus sebagai pengontrol yang mendukung koordinasi dan interaksi antara keduanya. *ViewModel* juga membantu mempertahankan *state* tampilan, memanipulasi *Model* yang dilakukan pada *View*, dan memicu peristiwa dalam *View* itu sendiri (Raj, Raman, & Subramanian, 2017).

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



Gambar 2.20 Rancangan arsitektur MVVM

Sumber: Arif, Musthafa, & Muriyatmoko (2019)

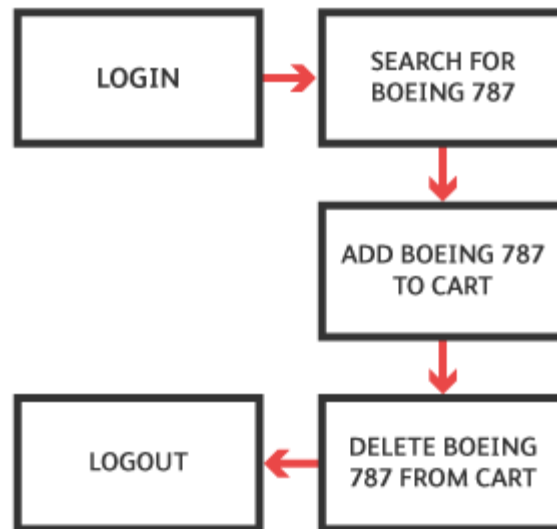
2.13 Pengujian Perangkat Lunak

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).

2.13.1 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 adalah untuk menemukan *bug* yang tidak akan dideteksi pengujian lain. Skenario di sini menyerupai instruksi terkait

bagaimana perangkat lunak digunakan, dan skenario pengujian dapat berupa kasus pengujian terpisah, atau serangkaian kasus-kasus pengujian yang saling terhubung (QATestLab, 2018).



Gambar 2.21 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).

Story: Upsell products.

In order to generate future sales,
As a tool-store owner,
I want to print out sale coupons for related items at checkout.

Scenario: Upsell a hammer for minimum threshold of nails sold.

Given a customer purchases nails,
And the amount of nails is over 2 boxes,
Then the customer receives a 20% coupon code for a hammer at checkout.

Gambar 2.22 Contoh sebuah *scenario-based testing*

Sumber: SoapUI (2022)

2.13.2 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).

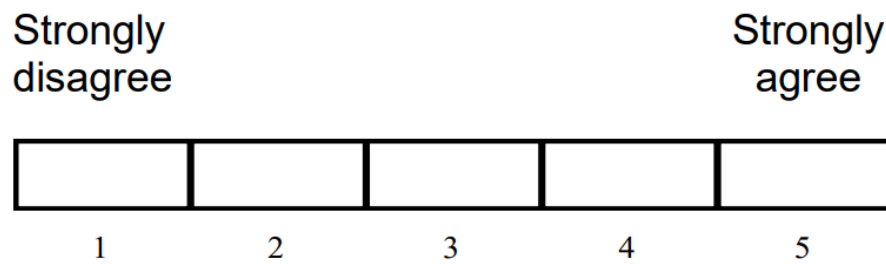
Salah satu teknik pengukuran dalam *usability testing* adalah *System Usability Scale (SUS)*. *System Usability Scale (SUS)* adalah sebuah teknik pengujian dengan cara melibatkan pengguna akhir dalam pengujiannya. Instrumen pengujian dalam *SUS* yaitu 10 pertanyaan sebagai tolak ukur pengujian, dan skala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

Berikut merupakan instrumen pengujian *System Usability Scale*.

Tabel 2.5 Instrumen pengujian *System Usability Scale*

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

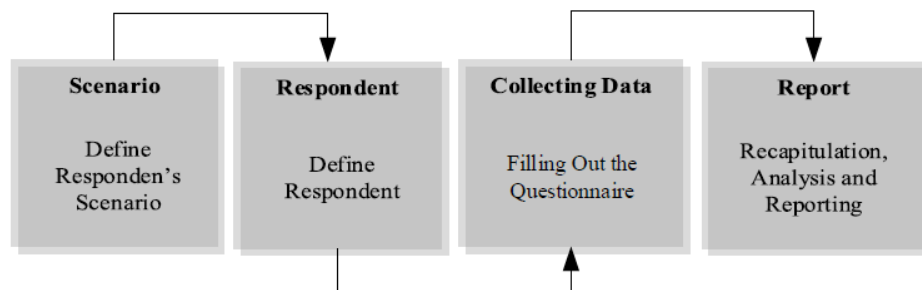
Sumber: Ependi, Kurniawan, & Panjaitan (2019)



Gambar 2.23 Skala penilaian *System Usability Scale*

Sumber: Brooke (1995)

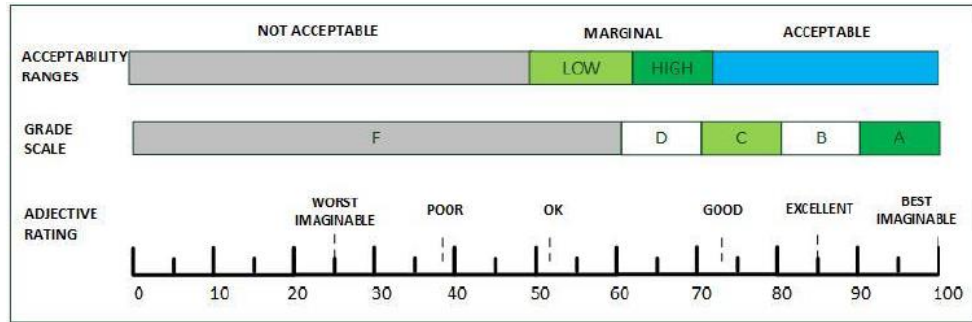
Langkah pengujian dalam *System Usability Scale* dimulai dari penentuan skenario, menentukan responden, melakukan pengujian oleh responden, dan merekapitulasi hasil pengujian.



Gambar 2.24 Langkah-langkah *System Usability Scale*

Sumber: Ependi, Kurniawan, & Panjaitan (2019)

Cara perhitungan hasil pengujian dalam *System Usability Scale* antara lain pada pertanyaan instrumen dengan nomor ganjil skala jawaban instrumen dikurangi satu, pada pertanyaan instrumen dengan nomor genap memiliki perhitungan angka lima dikurangi skala jawaban instrumen, hasil penilaian skala 0 sampai 4 dengan 4 sebagai jawaban terbaik, penjumlahan jawaban dikali dengan 2.5, dan menentukan nilai rata-rata jawaban instrumen pengujian semua responden. Dalam menentukan hasil perhitungan, terdapat 3 aspek yaitu *acceptability ranges*, *grade scale*, dan *adjective rating*. *Acceptability ranges* digunakan untuk mengetahui tingkat penerimaan pengguna terhadap perangkat lunak. *Grade scale* digunakan untuk mengetahui tingkatan perangkat lunak. *Adjective rating* digunakan untuk melihat rating dari perangkat lunak. (Ependi, Kurniawan, & Panjaitan, 2019).



Gambar 2.25 Penilaian *System Usability Scale*

Sumber: Ependi, Kurniawan, & Panjaitan (2019)

BAB 3 METODOLOGI

3.1 Tipe Penelitian

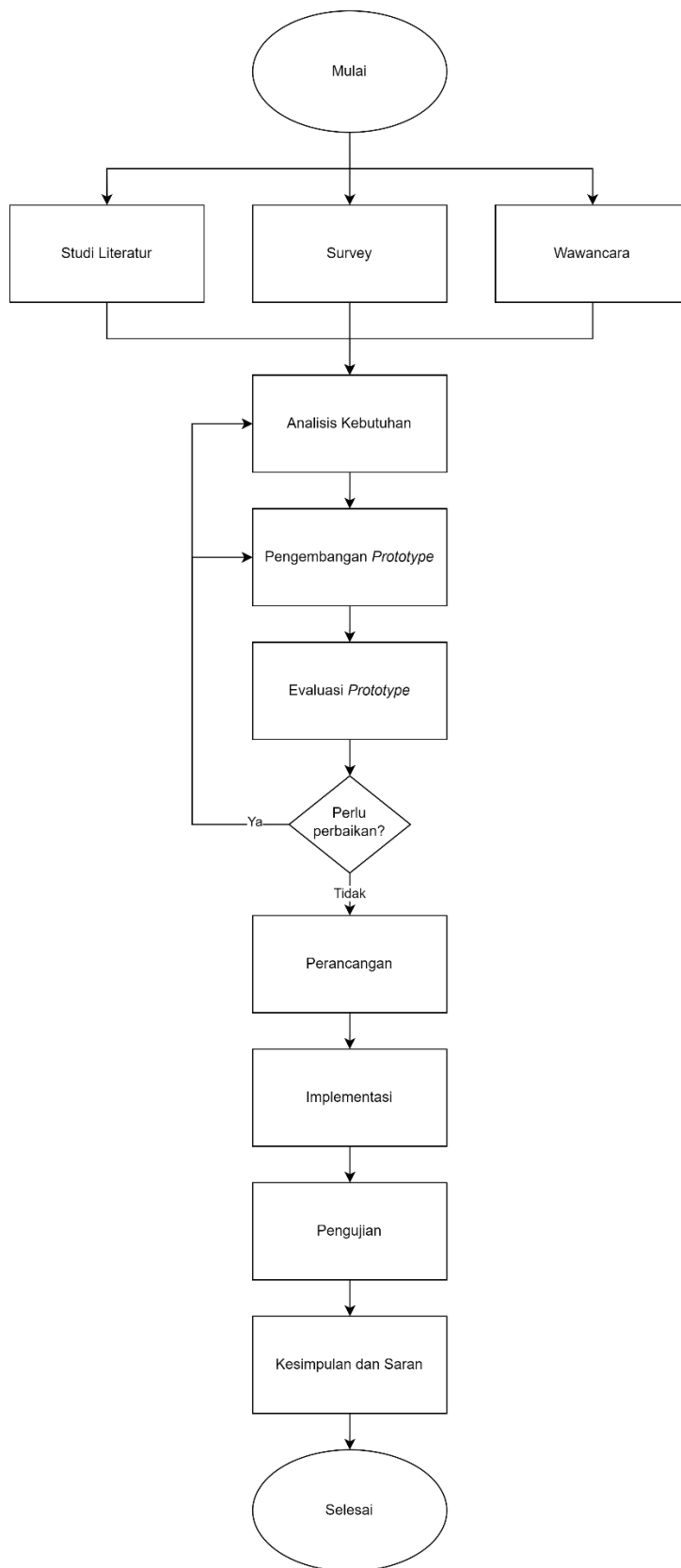
Tipe penelitian adalah sebuah klasifikasi dari jenis penelitian skripsi berdasarkan produk utama yang dihasilkan. Tipe penelitian yang diambil pada penelitian ini yaitu implementatif pengembangan. Hasil artefak dari penelitian ini yaitu sebuah aplikasi perangkat bergerak berbasis Android.

3.2 Metode Penelitian

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

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

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. Tahap Perancangan akan merancang perangkat lunak yang akan dikembangkan, berdasarkan *Prototype* yang telah tervalidasi pada tahap evaluasi. Tahap Implementasi akan mengimplementasikan rancangan perangkat lunak menjadi sebuah perangkat lunak. Perangkat lunak yang telah dibuat selanjutnya akan diuji pada tahap Pengujian, guna mengetahui tingkat ketepatan perangkat lunak yang dikembangkan dengan spesifikasi yang telah ditetapkan, dan sebagai bahan penarikan kesimpulan dan saran.



Gambar 3.1 Metode penelitian

3.3 Subjek Penelitian

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

3.4 Lokasi Penelitian

Lokasi penelitian ini dilakukan di Fakultas Ilmu Komputer Universitas Brawijaya.

3.5 Teknik Pengumpulan Data

Teknik pengumpulan data yang dipakai dalam penelitian “Pengembangan Aplikasi Perangkat Bergerak Panduan dan Resep Masakan untuk Mahasiswa Berbasis Android” adalah kuesioner via Google Forms, dan wawancara. Pengumpulan data kuesioner dilakukan 2 kali pada tahap pra-penelitian, yaitu kepada 130 masyarakat secara umum, dan secara spesifik kepada 82 mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya. Pada tahap analisis kebutuhan, akan dilakukan wawancara kepada mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya, untuk nantinya akan diobservasi untuk dijadikan acuan dalam pembuatan *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.

3.6 Peralatan Pendukung yang Digunakan

Peralatan pendukung yang digunakan pada pengembangan perangkat lunak dalam penelitian ini terdiri dari dua macam peralatan yaitu perangkat lunak dan perangkat keras. Perangkat lunak yang digunakan yaitu Android Studio untuk pembuatan aplikasi perangkat bergerak Android, dan Figma untuk pembuatan *Prototype* dan desain antarmuka perangkat lunak. Perangkat keras yang digunakan yaitu laptop ASUS f570zd, dan perangkat bergerak dengan sistem operasi Android 11.

3.7 Metode Pengembangan

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

3.7.1 Studi Literatur, Survey, Wawancara

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 kepastakaan. 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.

3.7.2 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 mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya untuk Analisis Kebutuhan.

Tabel 3.1 Daftar Pertanyaan Wawancara untuk Analisis Kebutuhan

No	Pertanyaan
1	Nama
2	Program Studi
3	Angkatan
4	Apakah anda suka memasak?
5	Makanan apa yang biasa anda masak?
6	Darimana anda belajar memasak makanan tersebut?
7	Apa saja permasalahan yang dihadapi saat memasak?
8	Menurut anda, bagaimana cara anda mengatasi permasalahan tersebut?

9	Apakah menu masakan yang terlalu rumit dapat mempersulit dalam memasak?
10	Apakah jika ada aplikasi yang memberikan resep masakan dan panduan memasaknya, anda akan terbantu dalam mengatasi permasalahan memasak yang mungkin terjadi?
11	Apa saja yang akan anda butuhkan dan ingin lakukan dari aplikasi resep tersebut?
12	Apakah ada kritik dan saran?

3.7.3 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*.

3.7.4 Evaluasi *Prototype*

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

Berikut merupakan daftar pertanyaan wawancara kepada mahasiswa 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 <i>Prototype</i> 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 <i>Prototype</i> ini?

3.7.5 Perancangan

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

3.7.6 Implementasi

Tahap implementasi akan mengimplementasikan rancangan perangkat lunak menjadi sebuah perangkat lunak. Proses implementasi dilakukan dengan mengembangkan sebuah perangkat lunak berupa aplikasi perangkat bergerak berbasis Android dengan menggunakan Android Studio menggunakan bahasa pemrograman Kotlin. Implementasi bertujuan untuk menghasilkan sebuah perangkat lunak yang dapat digunakan oleh pengguna untuk mencapai tujuannya.

3.7.7 Pengujian

Perangkat lunak yang telah dibuat selanjutnya akan diuji pada tahap Pengujian. Pengujian pada penelitian ini terdiri dari dua macam pengujian, yaitu *scenario-based testing*, dan *usability testing*. *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. *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 *System Usability Scale (SUS)*. *System Usability Scale (SUS)* adalah sebuah teknik pengujian dengan cara melibatkan pengguna akhir dalam pengujiannya. Instrumen pengujian dalam *SUS* yaitu 10 pertanyaan sebagai tolak ukur pengujian, dan skala penilaian sebagai ukuran pembobotan pengujian dari 1 hingga 5, dimana 1 menunjukkan sangat tidak setuju, dan 5 menunjukkan sangat setuju terhadap pertanyaan pengujian. Tahap pengujian bertujuan untuk mengetahui tingkat ketepatan perangkat lunak yang dikembangkan dengan spesifikasi yang telah ditetapkan, dan sebagai bahan penarikan kesimpulan dan saran.

Berikut merupakan daftar pertanyaan kuesioner kepada mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya untuk Pengujian.

Tabel 3.3 Daftar Pertanyaan Kuesioner untuk Pengujian

No	Pertanyaan	Skala
----	------------	-------

1	Saya ingin lebih sering menggunakan aplikasi ini	1 - 5
2	Saya merasa aplikasi ini tidak seharusnya dibuat serumit ini	1 - 5
3	Saya pikir aplikasi ini mudah digunakan	1 - 5
4	Saya membutuhkan bantuan dari orang teknis untuk menggunakan aplikasi ini	1 - 5
5	Saya menemukan bahwa fitur pada aplikasi ini terintegrasi dengan baik	1 - 5
6	Saya berfikir ada ketidaksesuaian dalam aplikasi ini	1 - 5
7	Saya merasa kebanyakan orang akan mudah mempelajari aplikasi ini dengan sangat cepat	1 - 5
8	Saya menemukan bahwa aplikasi ini sangat rumit untuk digunakan	1 - 5
9	Saya percaya diri untuk menggunakan aplikasi ini	1 - 5
10	Saya perlu belajar sebelum menggunakan aplikasi ini	1 - 5
11	Apakah ada kritik dan saran?	Isian

3.7.8 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.

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.uui.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/ijim/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 Manajemen 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 Manajemen 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. *Firestore Authentication | Simple, multi-platform sign-in*. [online] Tersedia di: <<https://firebase.google.com/products/auth>> [Diakses 9 Agustus 2022].
- Firebase, 2022c. *Get Started with Firestore 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 FirebaseAuth*. [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_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 Analisis 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 Manajemen 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-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_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]

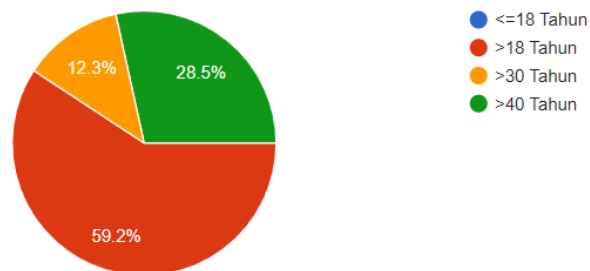
SoapUI, 2022. *Best Practices: Scenario-Based Testing*. [online] Tersedia di:
<<https://www.soapui.org/learn/functional-testing/scenario-based-testing/>> [Diakses 21 Agustus 2022]

LAMPIRAN A KUESIONER MASYARAKAT UMUM

A.1 Kuesioner Masyarakat Umum Tentang Resep Masakan

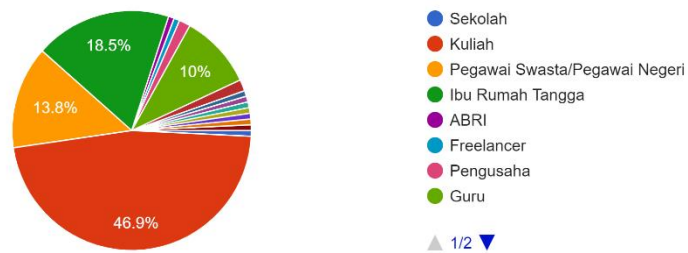
Usia

130 responses



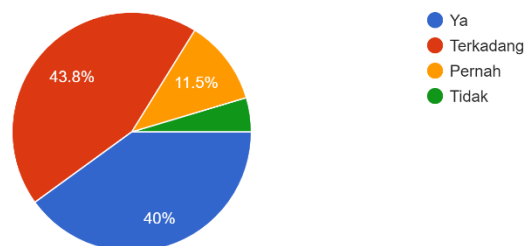
Pekerjaan

130 responses



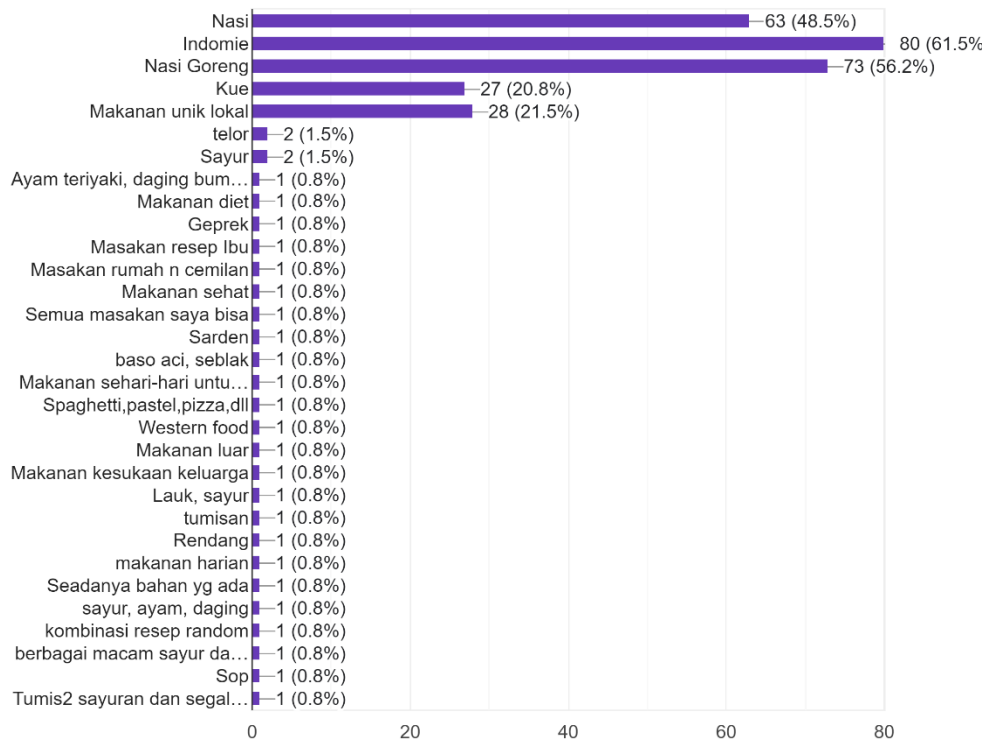
Apakah anda suka memasak?

130 responses



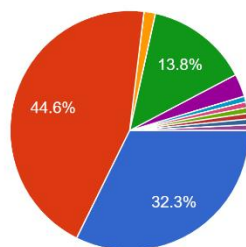
Makanan apa yang biasa anda masak?

130 responses



Darimana anda memasak makanan tersebut?

130 responses



- Diajarkan orang lain/Resep turun tem...
 - Dari resep/tutorial di internet
 - Dari resep/tutorial di buku resep
 - Membuat resep sendiri
 - Dimasakin oleh orang lain
 - Resep turun temurun kadang² liat inte...
 - Resep, tutorial
 - Resep dr tetangga
- ▲ 1/2 ▼

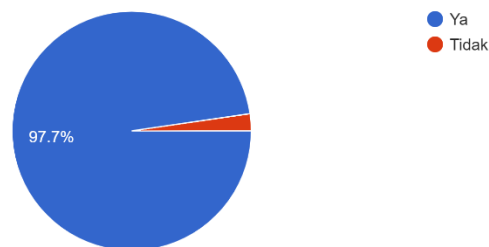
Apa saja permasalahan yang dihadapi saat memasak?

124 responses

- Tidak ada
- Mager
- Peralatan tidak memadai, kekurangan bahan
- susah nyari resep yg sesuai budget
- Bahan-bahan yang tidak biasa/asing namun ternyata ada nama lainnya yang lebih terdengar familiar
- Mengira-ngira takaran bumbu masakan, karena resep yang diajarkan orang lain/turun temurun biasanya tidak memiliki takaran pasti, atau sesuai selera saja/secukupnya.
- Antara terlalu matang dan masih mentah
- Sering lupa untuk memasukan bumbu
- Waktu memasak hingga benar-benar matang

Apakah jika ada aplikasi yang memberikan resep masakan, yang mana akan ada cara memasaknya juga, anda akan terbantu?

130 responses



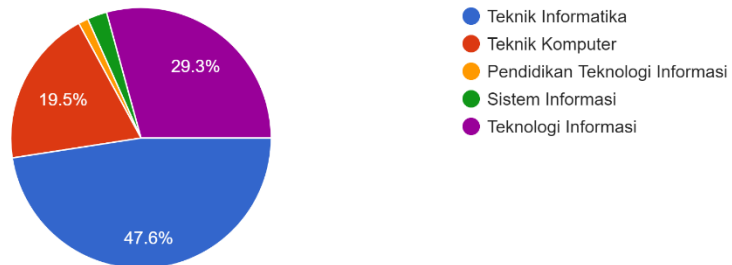
Fitur apa saja yang sekiranya dibutuhkan dalam aplikasi resep makanan?

130 responses

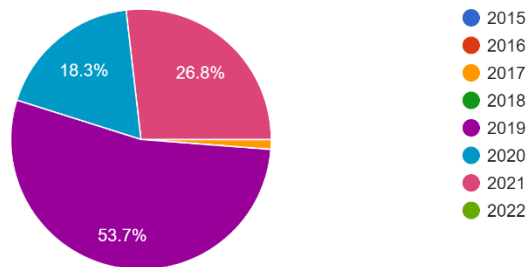
-
- Video
- langkah-langkah memasak
- Tahap tahap memasak, tips dan trik, beberapa contoh hasil masakan resep
- bisa nyari resep dg patokan harga bahan baku
- Fitur untuk menentukan takaran bahan-bahan yang dibutuhkan jika memasak untuk 1 porsi, 2 porsi atau bahkan banyak porsi
- Fitur rekomendasi masakan yang paling banyak dicari, tambahkan favorit, dan riwayat baca menu masakan
- Fitur langkah-langkah dalam membuat makanan

A.2 Kuesioner Masyarakat Umum Tentang Resep Masakan

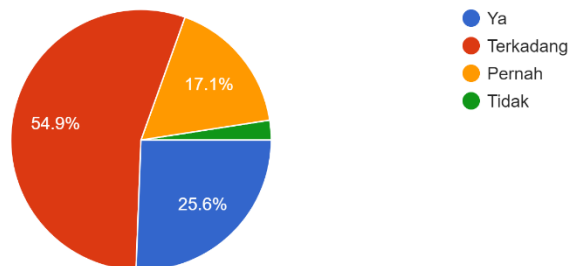
Program Studi
82 responses



Angkatan
82 responses

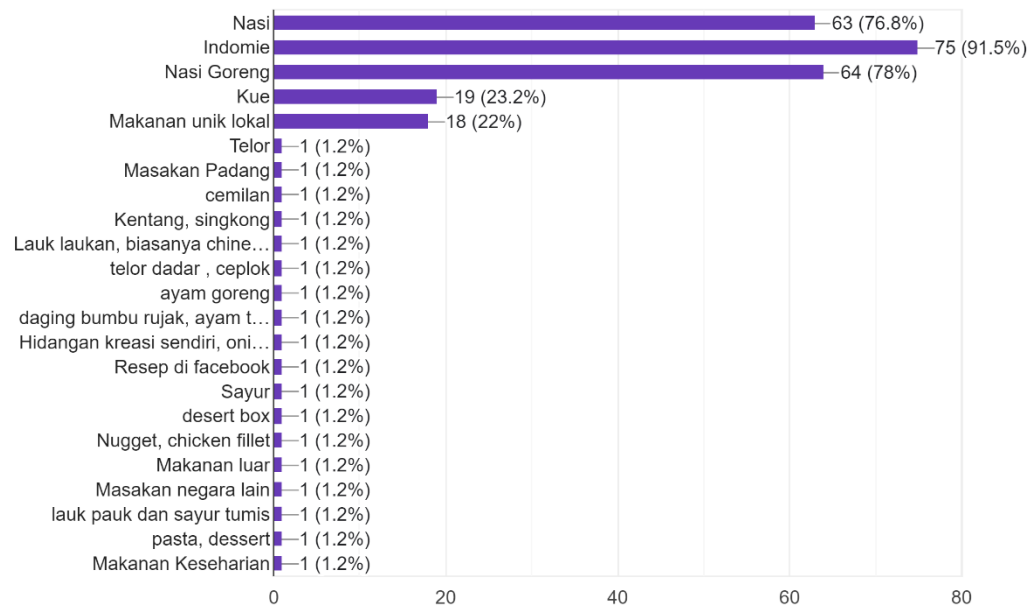


Apakah anda suka memasak?
82 responses



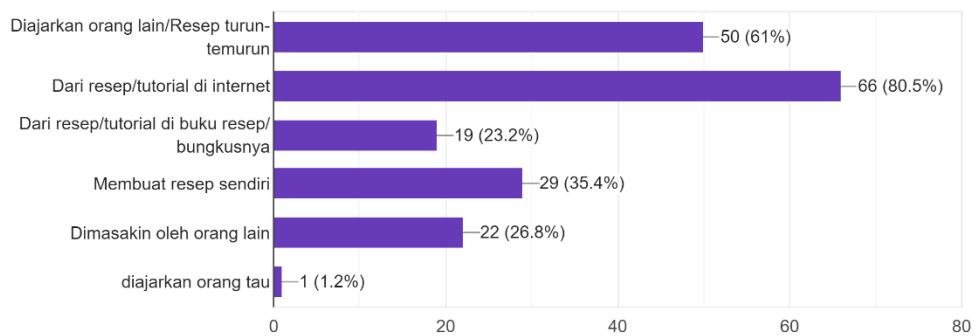
Makanan apa yang biasa anda masak?

82 responses



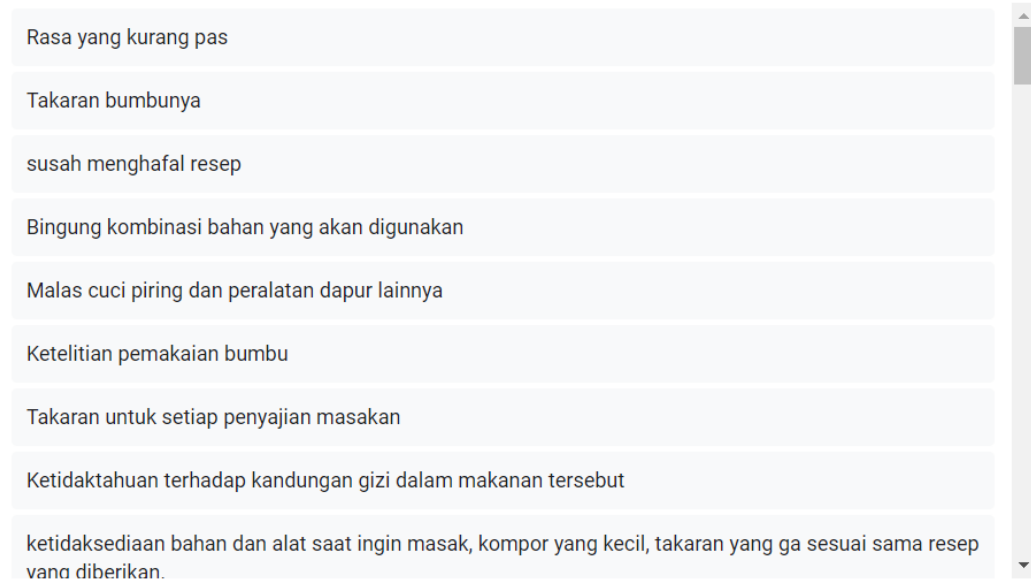
Darimana anda memasak makanan tersebut?

82 responses



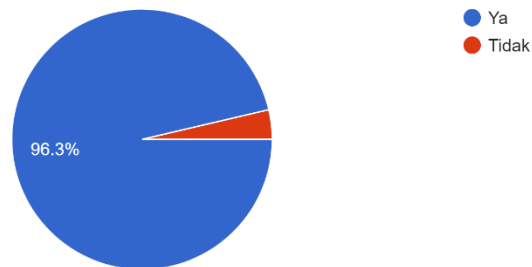
Apa saja permasalahan yang dihadapi saat memasak?

77 responses



Apakah jika ada aplikasi yang memberikan resep masakan, yang mana akan ada cara memasaknya juga, anda akan terbantu dalam mengatasi permasalahan memasak yang mungkin terjadi?

82 responses



Fitur apa saja yang sekiranya dibutuhkan dalam aplikasi resep makanan?

82 responses

Resep / Langkah langkah memasak makanan

Resep dan cara memasak

Step by step resep makanan secara mendetail

Video Tutorial

favorit untuk resep" kesukaan kita, ad fitur filter sesuai tema seperti gorengan, tumisan, dll

Timer dan ukuran bahan"

Takaran masakan per porsi, masakan untuk orang yang pemilih dalam hal makanan

Fitur yang dapat memberikan takaran yang pas dengan nutrisi hasil masakan

deskripsi alat dan bahan yang lengkap, terus juga langkah cara membuat sebuah makanan yang detail.