**Pengembang Framework untuk Membuat Aplikasi Seputar Permasalahan Gizi berbasis Platform Android**

**Proposal Tugas Akhir**

**PROPOSAL-SOFTWARE ENGINEERING [NKS]**

**1301198497**

**MUHAMMAD FAISAL AMIR**

****

**Program Studi Sarjana**

**Fakultas Informatika**

**Universitas Telkom**

**Bandung**

**2021**

# **LEMBAR PERSETUJUAN**

**Pengembang Framework untuk Membuat Aplikasi Seputar Permasalahan Gizi berbasis Platform Android**

**Framework developer to create applications around nutritional problems based on the Android platform**

**NIM: 1301198497**

**Muhammad Faisal Amir**

Proposal ini diajukan sebagai usulan pembuatan tugas akhir pada

Program Studi Sarjana

Fakultas Informatika Universitas Telkom

Bandung, 20 Mei 2021

Menyetujui

|  |  |  |
| --- | --- | --- |
| Calon Pembimbing 1  Dana Sulistyo Kusumo, S.T., M.T., PhD. |  | Calon Pembimbing 2  Shinta Yulia Puspitasari,  S.T, M.T  13880046 |

# **DAFTAR ISI**

[LEMBAR PERSETUJUAN 1](#_Toc75487157)

[DAFTAR ISI 2](#_Toc75487158)

[DAFTAR GAMBAR 3](#_Toc75487159)

[DAFTAR TABEL 4](#_Toc75487160)

[ABSTRAK 5](#_Toc75487161)

[1. PENDAHULUAN 6](#_Toc75487162)

[1.1. Latar Belakang 6](#_Toc75487163)

[1.2. Perumusan Masalah 7](#_Toc75487164)

[1.3. Tujuan 8](#_Toc75487165)

[1.4. Rencana Kegiatan 8](#_Toc75487166)

[1.5. Metodologi Penelitian 8](#_Toc75487167)

[1.6. Jadwal Kegiatan 10](#_Toc75487168)

[2. KAJIAN PUSTAKA 11](#_Toc75487169)

[2.1 Gizi 11](#_Toc75487170)

[2.2 Framework 11](#_Toc75487171)

[2.3 MVVM 12](#_Toc75487172)

[2.4 Android Arsitektur Component 13](#_Toc75487173)

[2.5 Design Thinking 15](#_Toc75487174)

[2.6 JitPack 16](#_Toc75487175)

[3. PERANCANGAN SISTEM 18](#_Toc75487176)

[3.1 Metodologi Penelitian 18](#_Toc75487177)

[3.1.1 Identifikasi Masalah 18](#_Toc75487178)

[3.1.2 Pengumpulan Data 20](#_Toc75487179)

[3.1.3 Perancangan Sistem 21](#_Toc75487180)

[3.1.4 Implementasi Sistem 22](#_Toc75487181)

[3.1.5 Pengujian Sistem 22](#_Toc75487182)

[3.1.6 Penarikan Kesimpulan dan Penulisan Laporan 22](#_Toc75487183)

[3.2 Perancangan Sistem 22](#_Toc75487184)

[DAFTAR PUSTAKA 29](#_Toc75487185)

[LAMPIRAN 31](#_Toc75487186)

# **DAFTAR GAMBAR**

[Gambar 2–1. Alur Kerja Framework CodeIgniter 11](#_Toc75498307)

[Gambar 2–2. Proses MVVM 13](#_Toc75498308)

[Gambar 2–3. Android Arsitektur Komponen 14](#_Toc75498309)

[Gambar 2–4. Design Thinking Step 15](#_Toc75498310)

[Gambar 2–5. Halaman Utama JitPack.io 17](#_Toc75498311)

[Gambar 3–1. Alur Metodologi Penelitian 18](#_Toc75498312)

[Gambar 3–2. Flowchart Pengguna 23](#_Toc75498313)

[Gambar 3–3. Flowchart Pengguna Akhir 24](#_Toc75498314)

[Gambar 3–4. Use Case Diagram Framework 25](#_Toc75498315)

[Gambar 3–5. Use Case Diagram Pengguna Akhir 27](#_Toc75498316)

**DAFTAR TABEL**

[Tabel 1‑1. Jadwal Kegiatan 10](#_Toc75497044)

[Tabel 3‑1. Tabel Skenario Use Case Framework 1 25](#_Toc75497045)

[Tabel 3‑2. Tabel Skenario Use Case Framework 2 25](#_Toc75497046)

[Tabel 3‑3. Tabel Skenario Use Case Framework 3 26](#_Toc75497047)

[Tabel 3‑4. Tabel Skenario Use Case Aplikasi 1 27](#_Toc75497048)

[Tabel 3‑5. Tabel Skenario Use Case Aplikasi 2 28](#_Toc75497049)

[Tabel 3‑6. Tabel Skenario Use Case Aplikasi 3 28](#_Toc75497050)

# **ABSTRAK**

Setiap individu memiliki karakteristik yang berbeda dalam aktivitas harian dan kesehatan. Pemahaman pengetahuan tentang gizi sangat diperlukan guna mencegah secara dini memburuknya kemungkinan yang akan terjadi. Pengembang android dituntut untuk mengembangkan aplikasi dengan cepat dan menghasilkan sumber kode yang ringkas, solusi yang bisa dilakukan yaitu dengan cara menerapkan design pattern dalam mengembangkan framework yang dapat mempersingkat waktu untuk pengembangan aplikasi. Adapun metode yang digunakan dalam melakukan penelitian mengenai Pengembang Framework untuk Membuat Aplikasi Seputar Permasalahan Gizi berbasis Platform Android, Identifikasi Masalah Mengumpulkan informasi dari beberapa jurnal ilmiah dan paper yang sudah membahas mengenai penelitian sebelumnya mengenai permasalahan gizi, framework. Mengumpulkan data yang mendukung untuk menentukkan kebutuhan fungsional, kebutuhan perangkat lunak dan perangkat keras yang digunakan untuk tahapan perancangan sistem. Merancang framework yang mudah digunakan untuk pengembangan aplikasi android dengan topik permasalahan seputar gizi. Setelah melakukan perancangan, maka dilanjutkan dengan tahap implementasi yaitu implementasi framework untuk aplikasi android dengan topik permasalahan seputar gizi. Implementasi sistem dikembangkan dengan menggunakan bahasa pemrograman kotlin. Hasil akhir dari implementasi ini adalah framework yang mudah digunakan pengembang android lain untuk mengembangkan aplikasi seputar permasalahan gizi. Pengujian sistem akan dilakukan setelah implementasi, dengan tujuan menguji tingkat efisiensi waktu dalam pembuatan aplikasi. Setelah pengujian selesai, dilakukan analisis terlebih dahulu sebelum akhirnya menarik kesimpulan dari sistem yang telah dibangun apakah framework ini mempermudah dalam pengembangan aplikasi android seputar permasalahan gizi, setelah melakukan penarikan kesimpulan, kemudian dilakukan tahap akhir pada penelitan yaitu membuat laporan secara keseluruhan untuk menyelesaikan penelitian.

# **PENDAHULUAN**

## **Latar Belakang**

Setiap individu memiliki karakteristik yang berbeda dalam hal jenis kelamin, usia, berat dan tinggi badan, aktivitas harian dan kesehatan [1]. Karakteristik tersebut menentukan kebutuhan zat gizi dan bahan pangan yang sebaiknya dikonsumsi [1] [2]. Terdapat 6 macam jenis zat gizi diantaranya yakni karbohidrat atau senyawa organik yang banyak terdapat pada tumbuhan dan binatang, lemak, protein, vitamin, mineral, dan air [2]. Rendahnya tingkat kecukupan zat gizi dapat mempengaruhi tingkat kecukupan energi pula, dampak kekurangan gizi dapat menyebabkan beberapa penyakit antara lain obesitas, pembengkakan hati dan ginjal, busung lapar, darah suka membeku, serta terhambatnya pertumbuhan [3].

Pemahaman pengetahuan tentang gizi dan upaya-upaya peningkatan perbaikan gizi sangat diperlukan guna mencegah secara dini memburuknya kemungkinan yang akan terjadi. Dengan adanya kendala waktu dan biaya yang tidak sedikit untuk konsultasi pada ahli gizi [4]. Para praktisi memanfaatkan teknologi digital dengan menggunakan perangkat bergerak untuk melakukan edukasi mengenai pentingnya keseimbangan zat gizi, saat ini belum banyak aplikasi berbasis smartphone android yang khusus memberikan layanan informasi gizi [1].

Google menyebutkan saat ini ada lebih dari tiga miliar pengguna aktif Android yang tersebar di seluruh dunia. Pengumuman ini disampaikan Sameer Samat selaku *Vice President Product Management Google* [5]. Di Indonesia android merupakan sistem operasi smartphone yang memiliki pangsa pasar terbesar [1] [6]. Berdasarkan data dari GlobalStats, tercatat dari bulan Mei 2020 sampai Mei 2021, 91.42% penduduk Indonesia menggunakan perangkat bergerak dengan sistem operasi android [7].

Pengembang android dituntut untuk mengembangkan aplikasi dengan cepat dan menghasilkan sumber kode yang ringkas, solusi yang bisa dilakukan yaitu dengan cara menerapkan *design pattern* dalam mengembangkan framework yang dapat mempersingkat waktu untuk pengembangan aplikasi [6].

Framework dapat didefinisikan sebagai kerangka aplikasi yang terdiri dari kode yang dikembangkan untuk semua fungsi dasar suatu sistem, yang dapat disesuaikan dalam mengembangkan aplikasi. Salah satu alasan utama mengembangkan framework adalah kode yang telah dibuat dapat digunakan kembali [8].

Di dalam sebuah framework terdapat pola arsitektur yang memiliki konsep dasar yaitu memisahkan logika bisnis dengan tampilan, jenis – jenis dari pola arsitektur untuk pengembangan aplikasi android ada 3, MVC (*Model View Controller*, MVP (*Model View Presenter*), MVVM (*Model View ViewModel*) [9]. Dari semua pola arsitektur tersebut, MVVM merupakan pola arsitektur terbaru dan terbaik untuk diterapkan dalam pengembangan aplikasi android, MVVM menggabungkan keuntungan dari pemisahan perhatian yang diberikan oleh MVP. Hasilnya adalah pola di mana model menjalankan operasi sebanyak mungkin, meminimalkan logika dalam tampilan [10]. Untuk dapat digunakan dalam setiap pengembangan aplikasi android, framework yang telah dirancang perlu untuk dipublikasikan, salah penyedia jasa untuk publikasi kumpulan framework yaitu JitPack.io [11].

Berdasarkan hal-hal yang telah disampaikan diatas penulis ingin membuat Pengembang Framework untuk Membuat Aplikasi Seputar Permasalahan Gizi berbasis Platform Android yang dapat dikembangkan oleh pengembang android secara mudah serta dapat digunakan oleh para pengguna untuk mengetahui seputar permasalahan gizi mereka.

## **Perumusan Masalah**

Dari latar belakang tersebut dapat dirumuskan masalah sebagai berikut:

1. Bagaimana mengembangkan Framework untuk membuat aplikasi seputar permasalahan gizi?
2. Bagaimana menerapkan Framework yang dapat di gunakan sesuai kebutuhan permasalahan user?

## **Tujuan**

Berdasarkan permasalahan yang telah diuraikan, maka penelitian ini memiliki tujuan sebagai berikut:

1. Dapat mengembangkan Framework untuk membuat aplikasi seputar permasalahan gizi.
2. Dapat menerapkan Framework yang dapat di gunakan sesuai kebutuhan permasalahan user.

## **Rencana Kegiatan**

Metode pengumpulan data adalah dengan cara mengkaji dari beberapa referensi sesuai dengan topik seperti Laporan dan Paper. Pada proses pembuatan Aplikasi Android menggunakan Aplikasi Android Studio.

## **Metodologi Penelitian**

Adapun metode yang digunakan dalam melakukan penelitian mengenai Pengembang Framework untuk Membuat Aplikasi Seputar Permasalahan Gizi berbasis Platform Android, yaitu:

1. Identifikasi Masalah

Mengumpulkan informasi dan referensi dari beberapa jurnal ilmiah dan paper yang sudah membahas mengenai penelitian sebelumnya mengenai permasalahan gizi, framework, dan arsitektur komponen android, pola arsitektur framework, kemudian memahami dan mengidentifkasi isi dari jurnal ilmiah dan paper sebagai cara untuk menemukan permasalahan yang dapat diangkat menjadi bahan penelitian. Proses pengidentifikasi permasalahan menggunakan metode *design thinking.*

1. Pengumpulan Data

Mengumpulkan data-data yang mendukung untuk menentukkan kebutuhan fungsional, kebutuhan perangkat lunak dan perangkat keras yang akan digunakan untuk tahapan perancangan sistem. Pengumpulan data ini dilakukan dengan cara studi literature dari referensi terkait.

1. Perancangan Sistem

Merancang framework yang mudah digunakan untuk pengembangan aplikasi android dengan topik permasalahan seputar gizi.

1. Implementasi Sistem

Setelah melakukan perancangan, maka dilanjutkan dengan tahap implementasi sistem yaitu implementasi framework untuk aplikasi android dengan topik permasalahan seputar gizi.

Implementasi sistem dikembangkan dengan menggunakan bahasa pemrograman kotlin. Hasil akhir dari implementasi ini adalah framework yang mudah digunakan pengembang android lain untuk mengembangkan aplikasi seputar permasalahan gizi.

1. Pengujian Sistem

Pengujian sistem akan dilakukan setelah implementasi, dengan tujuan untuk menguji tingkat efisiensi waktu dalam pembuatan aplikasi, serta membuktikan apakah rancangan sesuai dengan apa yang diimplementasikan.

Pengujian dilakukan dengan studi kasus pada programmer android pemula hingga tingkat ahli dengan cara mencoba framework hasil implementasi. Setelah pengujian akan dilakukan analisis lanjut untuk memperbaiki sistem yang telah dibuat.

1. Penarikan Kesimpulan dan Penulisan Laporan

Setelah pengujian selesai kemudian dilakukan analisis terlebih dahulu sebelum akhirnya menarik kesimpulan dari sistem yang telah dibangun apakah framework ini mempermudah dalam pengembangan aplikasi android seputar permasalahan gizi, setelah melakukan penarikan kesimpulan, kemudian dilakukan tahap akhir pada penelitan yaitu membuat laporan secara keseluruhan untuk menyelesaikan penelitian.

## **Jadwal Kegiatan**

Tabel 1‑1. Jadwal Kegiatan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Kegiatan** | **Bulan (2021)** | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Penentuan Topik |  |  |  |  |  |  |
| Penentuan Metode |  |  |  |  |  |  |
| Pengumpulan Data |  |  |  |  |  |  |
| Perancangan System |  |  |  |  |  |  |

# **KAJIAN PUSTAKA**

* 1. **Gizi**

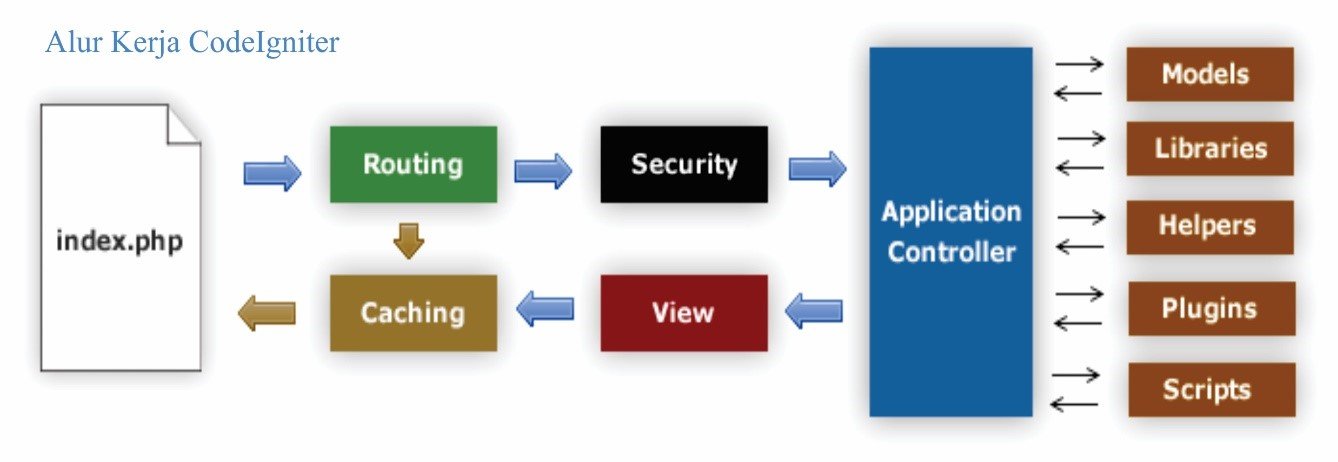
Gizi adalah aspek yang berhubungan dengan fungsi dasar zat gizi tersebut yaitu menghasilkan energi, pertumbuhan dan pemeliharaaan jaringan, serta mengatur proses metabolisme dalam tubuh [2]. Dapat juga di definisikan sebagai persediaan bahan-bahan atau makanan yang dibutuhkan organisme maupun sel-sel untuk bertahan hidup, sementara dalam bidang ilmu pengetahuan dan medis, gizi dapat merujuk pada ilmu atau praktik konsumsi serta penggunaan makanan [12].

Pengertian dari ilmu gizi adalah ilmu yang mempelajari segala sesuatu tentang makanan dalam hubungannya dengan kesehatan optimal. Ruang lingkup ilmu gizi adalah konsumsi pangan, cara-cara pemanfaatan makanan oleh tubuh baik dalam keadaan sehat maupun sakit. Sedangkan ilmu yang berkaitan dengan ilmu gizi adalah Biokimia, Biologi, Fisiologi, dan Patologi [2].

* 1. **Framework**

Framework (kerangka kerja) adalah istilah yang sering muncul dalam dunia developer. Istilah tersebut memiliki fungsi yang sangat besar bagi pengembangan kode program secara sistematis, Keuntungan lain adalah untuk mengembangkan perangkat lunak dengan penyusunan kode secara terstruktur dan konsisten. [13]

Kerangka kerja diciptakan untuk mempermudah kinerja dari programmer. Sehingga, seorang programmer tidak perlu untuk menuliskan kode secara berulang. Karena di dalamnya sendiri anda hanya perlu menyusun komponen – komponen pemrograman saja. [13] [14] [15].



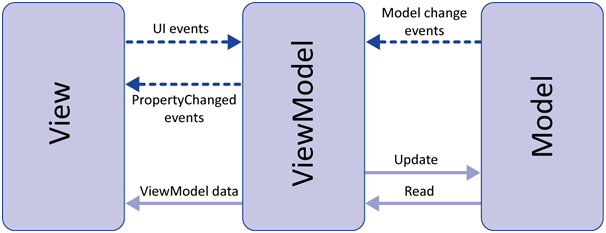
Gambar 2–1. Alur Kerja Framework CodeIgniter

Penulis menampilkan alur kerja framework codeigniter sebagai contoh. CodeIgniter adalah sebuah web application network yang bersifat *opensource* yang digunakan untuk membangun aplikasi php dinamis. [16]

CodeIgniter menjadi sebuah framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat [16]. Berikut penjelasan dari gambar alur kerja framework codeigniter:

1. Index.php: Index.php disini berfungsi sebagai file pertama dalam program yang akan dibaca oleh program [16].
2. The Router: Router akan memeriksa HTTP request untuk menentukan hal apa yang harus dilakukan oleh program [16].
3. Cache File: Apabila dalam program sudah terdapat “cache file” maka file tersebut akan langsung dikirim ke browser. File cache inilah yang dapat membuat sebuah website dapat di buka dengan lebih cepat. Cache file dapat melewati proses yang sebenarnya harus dilakukan oleh program codeigniter [16].
4. Security: Sebelum file controller di load keseluruhan, HTTP request dan data yang disubmit oleh user akan disaring terlebih dahulu melalui fasilitas security yang dimiliki oleh codeigniter [16].
5. Controller: Controller akan membuka file model, core libraries, helper dan semua resources yang dibutuhkan dalam program tersebut [16].
6. View: Hal yang terakhir akan dilakukan adalah membaca semua program yang ada dalam view file dan mengirimkannya ke browser supaya dapat dilihat. Apabila file view sudah ada yang di “cache” maka file view baru yang belum ter-cache akan mengupdate file view yang sudah ada [16].
   1. **MVVM**

MVVM dijelaskan oleh John Gossman dalam blognya. Arsitektur ini kemudian diterapkan dalam Microsoft Silverlight dan WPF. MVVM adalah varian modern dari Model-View-Controller (MVC) dan tujuan intinya adalah memisahkan antara komponen Model dan komponen View [14].

****

Gambar 2–2. Proses MVVM

1. Model

Komponen ini mewakili logika bisnis dan data. Hal tersebut berarti bahwa logika bisnis menentukan bagaimana agar data yang akan dimanipulasi terdapat di dalam komponen Model.

1. View

Komponen ini mewakili komponen antarmuka dan pada dasarnya akan berisi komponen antarmuka, seperti layout XML dsb.

1. ViewModel

ViewModel adalah komponen yang sangat penting dari arsitektur karena membantu dalam pemisahan presentasi, yaitu membantu untuk menjaga View terpisah dari Model dan pada saat yang sama, bertindak sebagai pengontrol yang mendukung interaksi dan koordinasi antara View dan Model. Komponen ViewModel juga berisi perintah dan metode yang membantu mempertahankan keadaan tampilan dan membantu memanipulasi Model sesuai tindakan, yang dilakukan pada View. Komponen ViewModel juga membantu memicu peristiwa dalam komponen View itu sendiri. [14]

* 1. **Android Arsitektur Component**

Komponen arsitektur Android adalah kumpulan library untuk membantu Anda mendesain aplikasi yang tangguh, dapat diuji, dan mudah pemeliharaannya. Dimulai dari class untuk mengelola siklus proses komponen UI dan menangani persistensi data. [17]

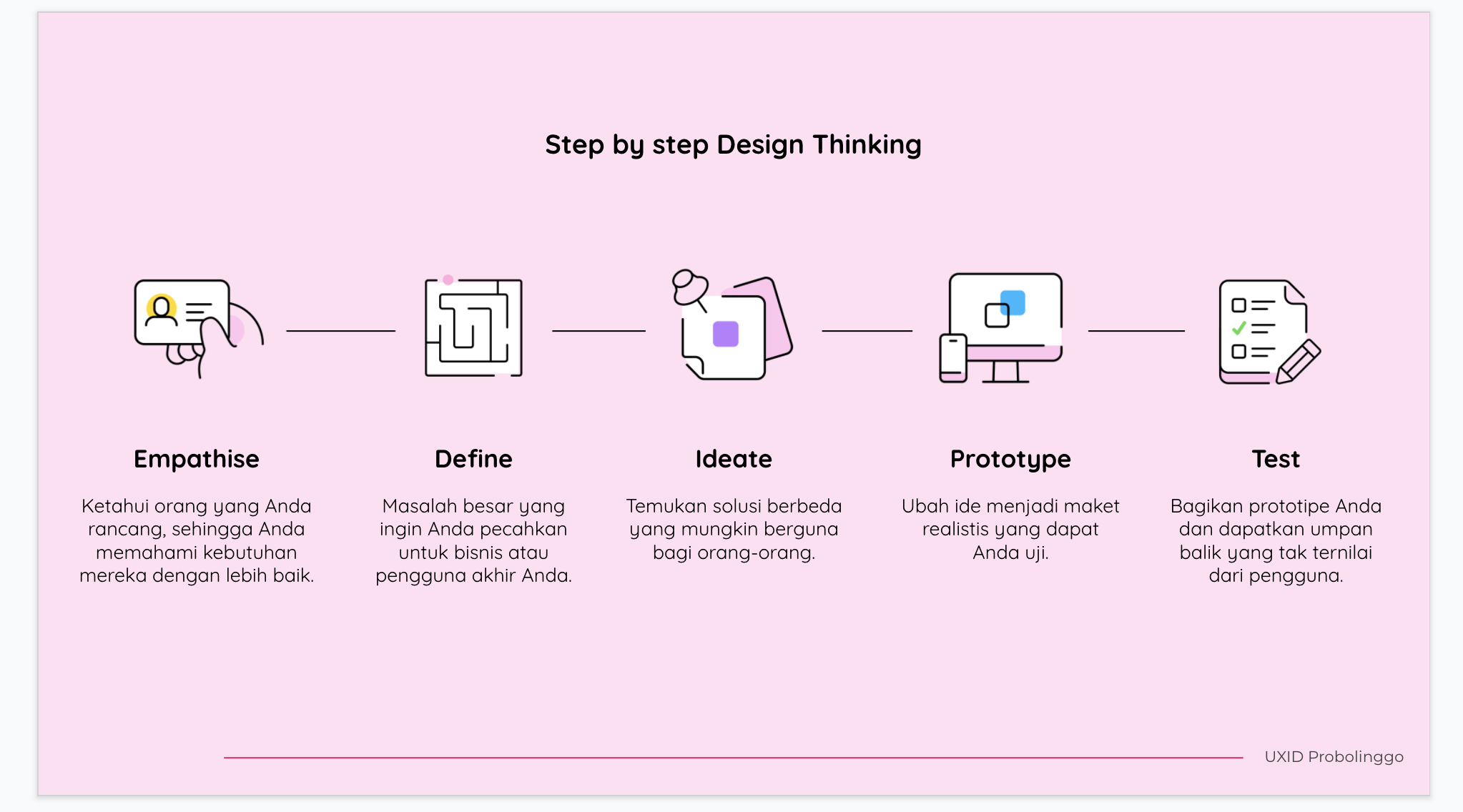
Untuk memahami apa yang tim Android usulkan, kita harus mengetahui semua elemen Komponen Arsitektur, karena merekalah yang akan melakukan pengangkatan berat bagi kita. Ada empat komponen, masing-masing dengan peran spesifik Room, ViewModel, LiveData, dan Lifecycle. Semua bagian itu ada tanggung jawab sendiri, dan mereka bekerja sama untuk menciptakan arsitektur yang kokoh. [17] [18]



Gambar 2–3. Android Arsitektur Komponen

1. Activity dan Fragment mewakili View layer, yang tidak berhubungan dengan logika bisnis dan operasi yang kompleks. Ia hanya mengonfigurasi tampilan, menangani interaksi pengguna, dan yang terpenting, pengamatan dan menunjukkan elemen LiveData yang diambil dari ViewModel [14].
2. ViewModel secara otomatis mengamati keadaan tampilan Lifecycle, menjaga konsistensi selama perubahan konfigurasi dan lifecycle events Android lainnya. Hal ini juga dituntut oleh tampilan untuk mengambil data dari Repository, yang disediakan sebagai LiveData yang dapat diamati. Penting untuk dipahami bahwa ViewModel tak pernah merujuk View secara langsung dan update data selalu dilakukan oleh LiveData kesatuan [18].
3. Repository bukan komponen Android khusus. Ia adalah kelas sederhana, tanpa implementasi tertentu, yang bertanggung jawab untuk mengambil data dari semua sumber yang tersedia, dari database ke layanan web. Ia menangani semua data ini, umumnya mengubahnya menjadi LiveData yang tampak dan membuat mereka tersedia untuk ViewModel [18].
4. Database Room adalah perpustakaan pemetaan SQLite yang memfasilitasi proses berurusan dengan database. Secara otomatis ia menulis satu ton boilerplate, memeriksa kesalahan pada waktu kompilasi, dan yang terbaik, ia dapat langsung mengembalikan kueri dengan LiveData yang tampak [18].
   1. **Design Thinking**

*Design Thinking* adalah proses berulang dimana kita berusaha memahami pengguna, menantang asumsi, dan mendefinisikan kembali masalah dalam upaya mengidentifikasi strategi dan solusi alternatif yang mungkin tidak langsung terlihat dengan tingkat awal pemahaman kita. Pada saat yang sama, Design Thinking menyediakan pendekatan berbasis solusi untuk menyelesaikan masalah. Ini adalah cara berpikir dan bekerja serta kumpulan metode langsung.



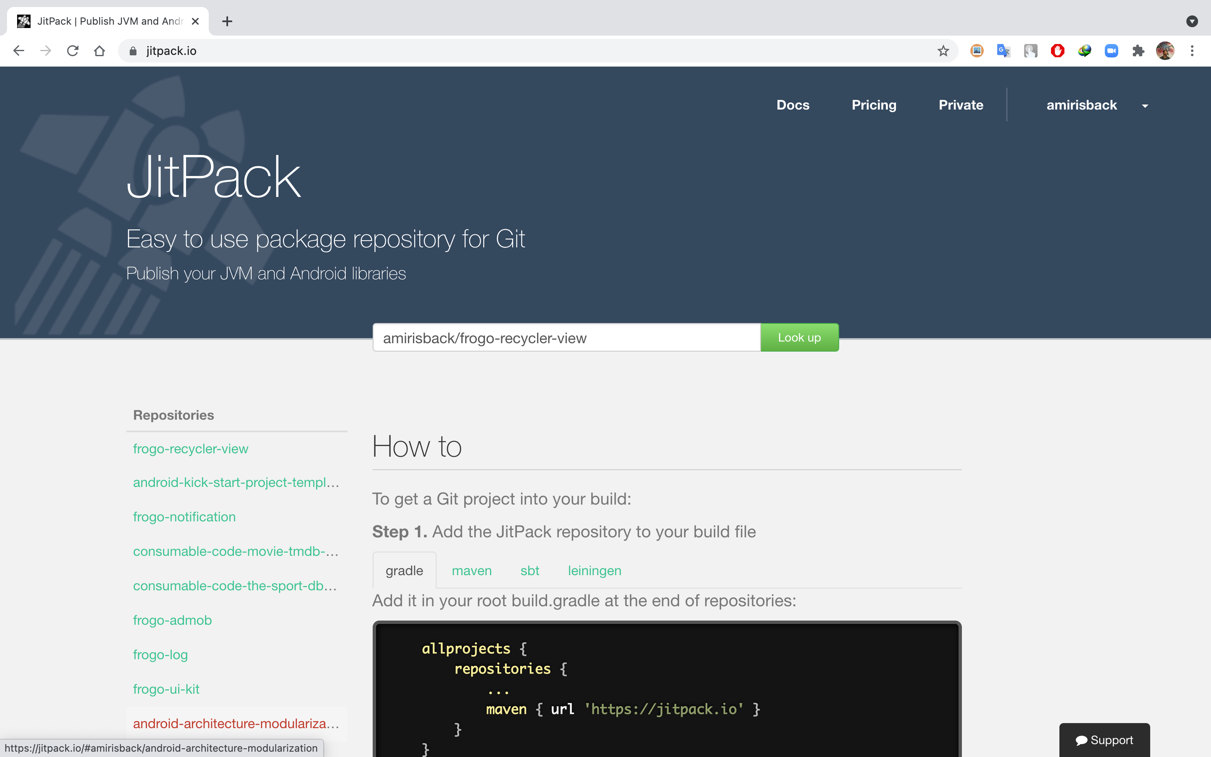
Gambar 2–4. Design Thinking Step

*Design Thinking* sangat berguna dalam mengatasi masalah-masalah yang tidak jelas atau tidak dikenal, dengan melakukan reframing masalah dengan cara-cara yang berpusat pada manusia, menciptakan banyak ide dalam brainstorming, dan mengadopsi pendekatan langsung dalam pembuatan prototype dan testing.

Design Thinking juga melibatkan eksperimen yang sedang berjalan: membuat sketsa, membuat prototype, testing, dan mencoba berbagai konsep dan ide. Berikut tahapan dalam proses *Design Thinking*:

1. *Empathise*, Tahap ini berfungsi untuk mendapatkan pemahaman empatik tentang masalah yang dicoba untuk diselesaikan*. Design Thinking*, dan empati memungkinkan pemikir desain untuk mengesampingkan asumsi mereka sendiri tentang dunia untuk mendapatkan wawasan tentang pengguna dan kebutuhan mereka [19].
2. Define, kita mengumpulkan informasi yang telah kita buat dan kumpulkan selama tahap *Empathise*, memungkinkan pengguna untuk menyelesaikan masalah sendiri dengan tingkat kesulitan minimal [19].
3. *Ideate*, desainer siap untuk mulai menghasilkan ide, Ada ratusan teknik *Ideation* seperti *Brainstorm*, *Brainwrite*, *Worst* *Possible* *Idea*, dan *SCRAMPER*, penting untuk mendapatkan sebanyak mungkin ide atau solusi masalah [19].
4. *Prototype*, Ini adalah fase eksperimental, dan tujuannya adalah untuk mengidentifikasi solusi terbaik untuk setiap masalah yang diidentifikasi selama tiga tahap pertama. Solusi diimplementasikan dalam prototype, dan satu per satu, mereka diselidiki dan diterima, diperbaiki dan diperiksa ulang, dan ditolak berdasarkan pengalaman pengguna [19].
5. Test, Desainer menguji produk lengkap secara ketat menggunakan solusi terbaik yang diidentifikasi selama fase prototyping. Ini adalah tahap akhir dari design thinking [19].
   1. **JitPack**

JitPack adalah repositori paket baru untuk proyek JVM dan Android. Itu membangun proyek Git sesuai permintaan dan memberi pembaca artefak siap pakai (jar, aar) [11].



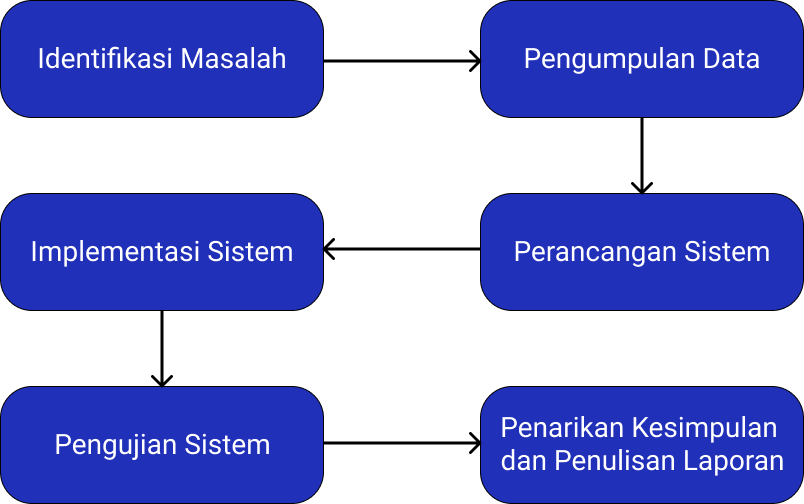
Gambar 2–5. Halaman Utama JitPack.io

Jika pembaca ingin perpustakaan pembaca tersedia untuk dunia, tidak perlu melalui langkah-langkah pembuatan dan pengunggahan proyek. Yang perlu pembaca lakukan adalah mendorong proyek pembaca ke GitHub dan JitPack akan mengurus sisanya. Jika proyek pembaca sudah ada di GitHub, JitPack memastikannya dapat dibangun oleh siapa saja [11].

# **PERANCANGAN SISTEM**

## **3.1 Metodologi Penelitian**

Tahap ini sudah disinggung pada bagian sebelumnya, akan tetapi pada bagian ini setiap tahapan metodologi penelitian akan diperjelas dan dilengkapi dengan flowchart. Berikut metodologi penelitian yang digunakan dalam menyelesaikan penelitian mengenai Pengembang Framework untuk Membuat Aplikasi Seputar Permasalahan Gizi berbasis Platform Android.



Gambar 3–1. Alur Metodologi Penelitian

### **3.1.1 Identifikasi Masalah**

Tahap ini dimulai dengan mengumpulkan referensi terkait mengenai pengembangan framework aplikasi android dan permasalahan gizi. Setelah pengumpulan referensi tahap selanjutnya adalah memahami dan mengidentifikasi referensi untuk menentukkan latar belakang yang akan digunakan pada perumusan masalah untuk mengangkat permasalahan dalam penelitian. Permasalahan yang diangkat didapatkan dari topik yang diberikan oleh dosen pembimbing 2 "Framework Untuk Pengembangan Permasalahan Gizi"

Berikut merupakan deskripsi dan pemodelan permasalahan dalam Pengembang Framework untuk Membuat Aplikasi Seputar Permasalahan Gizi berbasis Platform Android:

1. Identifikasi Persoalan

Proses pengidentifikasi permasalahan menggunakan metode *design thinking* dalam perumusan masalah, penulis mensortir dengan memilih 3 tahap. Empathise, Define, dan Ideate.

* 1. Empathise

Penulis membagi dalam 2 pengguna karena menurut empati dari penulis ada 2 pengguna yang membutuhkan aplikasi ini. Pertama ada pengguna biasa dan kedua ada pengguna akhir.

Untuk pengguna terdiri dari programmer, ahli gizi, dokter. Pengguna programmer memanfaatkan dalam pengembangan framework, sedangkan pengguna ahli gizi, dan dokter memanfaatkan sisi kustomasi programnya. Untuk pengguna akhir terdiri dari pasien dan pengguna yang mengalami masalah dengan gizi.

* 1. Define

Berikut masalah yang dialami oleh ke dua pengguna yang telah didefinisikan oleh penulis pada tahap empathise:

1. Pengguna
2. Pengembang yang membutuhan framework untuk customize dengan mudah aplikasi nutrisi.
3. Pengembang yang ingin mengembangkan aplikasi yang mudah di customize khususnya tentang nutrisi.
4. Ahli Nutrisi / dokter / ahli gizi yang ingin membantu end user dengan memberikan ilmunya.
5. Pengembang yang ingin mengembangkan aplikasi yang mudah di customize khususnya tentang nutrisi.
6. Pengguna Akhir
7. Orang yang memiliki masalah dengan nutrisi.
8. Orang yang memerlukan perhatian khusus dengan nutrisi mereka.
9. Berbagai macam masalah mengenai nutrisi.
10. Self Assesment (Penyakit bawaan / tidak, profiling orang untuk aplikasi.
    1. Ideate

Berikut solusi yang ditawarkan penulis atas permasalahan masing masing pengguna:

1. Pengguna Progammer

Membuat framework aplikasi android yang mudah untuk di customize oleh programmer, pengembang, bahkan oleh dokter dan ahli nutrisi

1. Pengguna Akhir

Membuat Aplikasi Android, karena jaman sekarang, android sudah menjadi gadget kebanyakan orang, (perlu data)  
Rekomendasi makanan / obat, rekomendasi penanganan

1. Identifikasi Pengguna

Pengguna dikelompokkan menjadi level penggunanya, yakni pengguna dan pengguna akhir, yang mempuyai fungsi dan hak yang berbeda beda terdiri dari:

1. Pengguna Progammer

Pengguna Progammer memiliki fungsi untuk kustomasi framework aplikasi sesuai dengan yang akan di rancang guna memenuhi kebutuhan gizi dari pengguna akhir.

1. Pengguna Akhir

Pengguna akhir memiliki fungsi untuk menggunakan aplikasi sesuai kebutuhan yang telah didefinisikan / dirancang oleh pengguna.

### **3.1.2 Pengumpulan Data**

Mengumpulkan data-data yang mendukung untuk menentukkan kebutuhan fungsional, kebutuhan perangkat lunak dan perangkat keras yang akan digunakan untuk tahapan perancangan sistem. Pengumpulan data ini dilakukan dengan cara studi literatur dari referensi terkait. Berikut keluaran dari hasil pengumpulan data untuk pembangunan sistem.

Tahap ini menjelaskan mengenai kebutuhan sistem dalam proses pengembangan yang terdiri dari beberapa kebutuhan yaitu:

1. Kebutuhan Fungsionalitas
2. Framework dapat digunakan untuk pengembangan aplikasi android seputar permasalahan gizi.
3. Framework dapat digunakan sebagai template dalam pengembangan aplikasi android seputar permasalahan gizi.
4. Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam penelitian dan pengembangan framework ini memiliki spesifikasi sebagai berikut:

* 1. Laptop / PC dengan spesifikasi
     1. Prosesor *Intel Core i5-7200U dual-core 2,5GHz TurboBoost 3,1GHz*
     2. *Grafis Intel HD Graphics 620 dan Nvidia GeForce GT 920MX VRAM 2GB*
     3. Memori RAM 8GB DDR4
     4. Konektifitas *LAN*, *WiFi*, *Port* USB 3.0
  2. Smartphone dengan spesifikasi
     1. Android API minimal 21
     2. RAM 2 GB
     3. Konetifitas Internet

1. Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian dan pengembangan framework ini memiliki spesifikasi sebagai berikut:

1. *IDE (Integrated Development Environment*) *Android Studio* v4.1.3
2. *Java Development Kit* 8
3. *Figma*

### **3.1.3 Perancangan Sistem**

Tahap ini berisikan tentang perancangan yang dibuat berdasarkan data yang sudah dikumpulkan pada tahap sebelumnya. Data tersebut digunakan untuk pengembangan framework aplikasi android seputar permasalahan gizi. Dalam perancangan ini berisikan tentang bagaimana gambaran sistem yang akan dibangun, dan menjelaskan cara kerja algoritma yang diterapkan. Tahap ini juga membuat skenario pengujian untuk sistem yang dikembangkan. Output dari perancangan sistem ini adalah rancangan sistem yang terdiri dari arsitektur sistem dan skenario pengujian. Tahapan ini akan dijelaskan lebih lanjut pada bagian selanjutnya berisikan gambaran sistem dan alur kerja sistem yang dikembangkan.

### **3.1.4 Implementasi Sistem**

Tahap ini mengimplementasikan sistem yang sudah dirancang pada tahap sebelumnya. Implementasi disini merupakan tahap pengembangan sistem dimana kebutuhan sistem sudah didefinisikan pada tahap sebelumnya. Bahasa pemrograman yang digunakan pada tahap ini adalah kotlin. Implementasi didukung juga oleh JitPack sebagai tempat publikasi framework agar bisa digunakan.

### **3.1.5 Pengujian Sistem**

Pengujian adalah tahapan setelah tahap implementasi sistem. Pengujian sistem dilakukan dengan menggunakan skenario pengujian yang sudah dirancang pada tahap perancangan sistem. Output dari pengujian ini adalah hasil uji yang akan dijadikan fakta untuk membuktikan hipotesis pengujian dengan hasil sistem yang dikembangkan. Pada bagian ini menjelaskan tentang skenario pengujian terhadap sistem yang dibangun, untuk membuktikan sistem yang dibangun mampu memberikan konteks yang sesuai dengan kondisi situasional user.

### **3.1.6 Penarikan Kesimpulan dan Penulisan Laporan**

Tahap ini merupakan tahapan akhir dari metodologi penilitian yang direncanakan. Setelah melakukan beberapa tahapan dalam metode penilitian, akan dilakukan analisis terlebih dahulu terhadapt hasil dari pengujian sistem, kemudian setelah melakukan analisis akan dilakukan penarikan kesimpulan berdasarkan sistem yang diuji pada tahap sebelumnya. Kesimpulan yang didapatkan dari hasil analisis dapat menghasilkan suatu saran untuk yang mungkin bisa digunakan untuk tahapan penelitian selanjutnya dari penelitian yang sudah dilakukan

## **3.2 Perancangan Sistem**

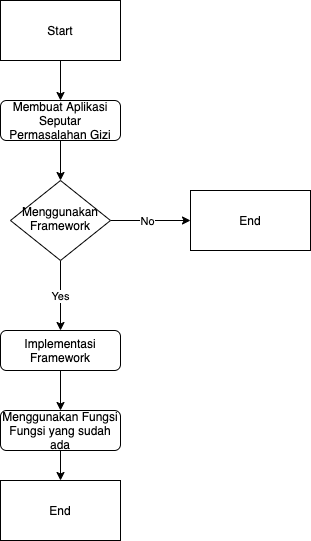
Tahap ini menjelaskan tentang perancangan sistem yang akan dibangun dimulai dari alur sistem bekerja, kemudian penjelasan alur kerja metode yang digunakan dalam pengembangan sistem dan berisikan tentang use case dari sistem beserta dengan spesifikasi use case yang dibuat.

* + 1. **Flowchart**

Berikut flowchart alur dari sistem berdasarkan pengguna yang telah di bahas diatas:

1. Pengguna Progammer

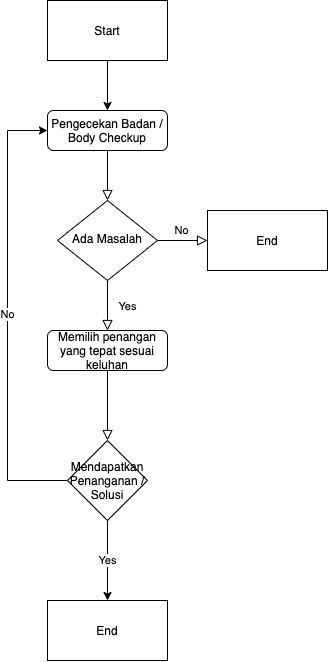
Pengguna Progammer memiliki fungsi untuk kustomasi framework aplikasi sesuai dengan yang akan di rancang guna memenuhi kebutuhan gizi dari pengguna akhir.



Gambar 3–2. Flowchart Pengguna

1. Pengguna Akhir

Pengguna akhir memiliki fungsi untuk menggunakan aplikasi sesuai kebutuhan yang telah didefinisikan / dirancang oleh pengguna.



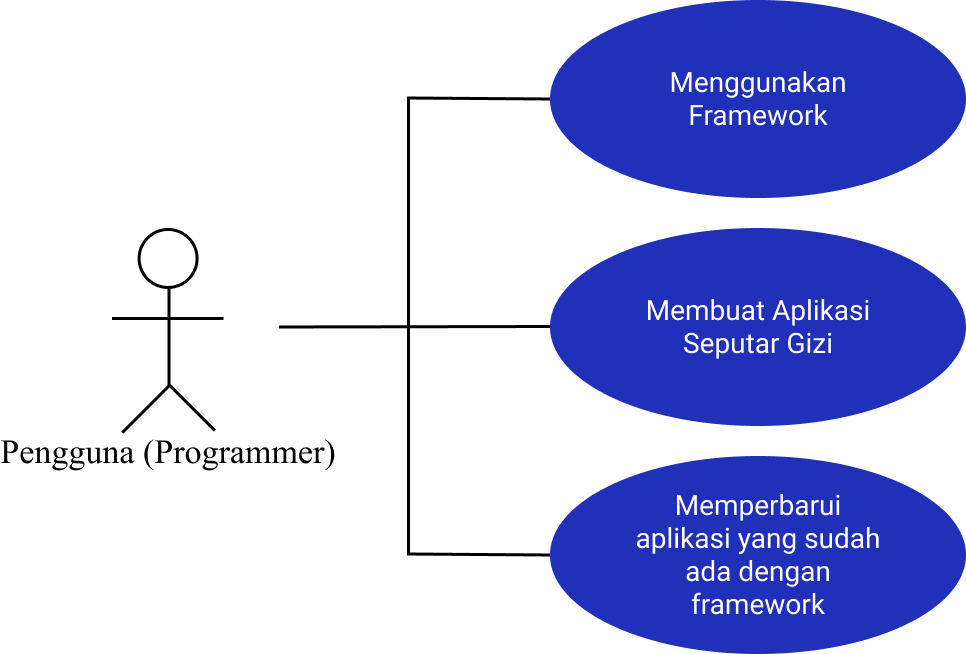
Gambar 3–3. Flowchart Pengguna Akhir

* + 1. **Use Case Diagram**

Berikut use case diagram dari sistem berdasarkan pengguna yang telah di bahas diatas:

1. Framework

Berikut *use case* diagram beserta skenario *use case* dari framework:



Gambar 3–4. Use Case Diagram Framework

Skenario Use Case dari Framework:

Tabel 3‑1. Tabel Skenario Use Case Framework 1

|  |  |  |  |
| --- | --- | --- | --- |
| No. | | 1 | |
| Nama Use Case | | Menggunakan Framework | |
| Aktor | | Pengguna (Programmer) | |
| Deskripsi | | Menggunakan framework untuk mengembangkan aplikasi android | |
| Pre-condition | | Aktor ingin menggunakan framework | |
| Post-condition | | Aktor telah menggunakan framework | |
| **Skenario Normal** | | | |
| No. | Aksi Aktor | | Reaksi Sistem |
| 1. | Pengguna menggunakan framework | |  |
| 2. |  | | Sistem merespon dengan cara menampilkan semua fungsi yang telah tersedia pada framework |
| 3. | Pengguna menggunakan fungsi framework sesuai kebutuhan | |  |
| 4. |  | | Sistem merespon dan segera melakukan perintah pengguna |

Tabel 3‑2. Tabel Skenario Use Case Framework 2

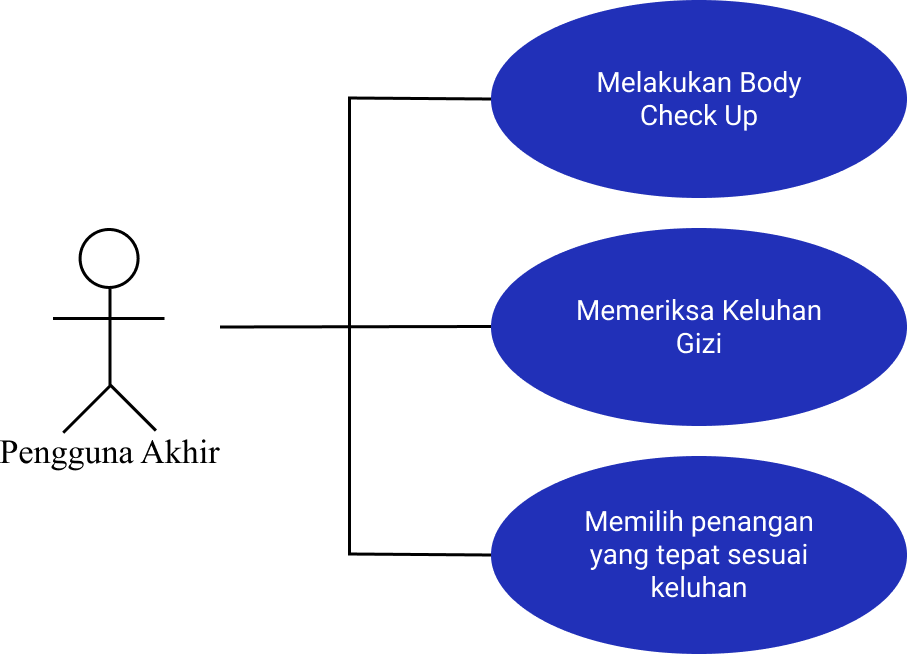
|  |  |  |  |
| --- | --- | --- | --- |
| No. | | 2 | |
| Nama Use Case | | Membuat Aplikasi Seputar Gizi | |
| Aktor | | Pengguna (Programmer) | |
| Deskripsi | | Membuat aplikasi seputar gizi dengan menggunakan framework sistem | |
| Pre-condition | | Aktor ingin membuat aplikasi seputar gizi | |
| Post-condition | | Aktor telah membuat aplikasi seputar gizi | |
| **Skenario Normal** | | | |
| No. | Aksi Aktor | | Reaksi Sistem |
| 1. | Pengguna membuat aplikasi seputar gizi menggunakan framework | |  |
| 2. |  | | Framework menampilkan semua fungsi yang tersedia |
| 3. | Pengguna memilih fungsi yang ditampilkan sistem | |  |
| 4. |  | | Framework bekerja menjalankan fungsi yang telah dipanggil |

Tabel 3‑3. Tabel Skenario Use Case Framework 3

|  |  |  |  |
| --- | --- | --- | --- |
| No. | | 3 | |
| Nama Use Case | | Memperbarui aplikasi yang sudah ada dengan menggunakan framework | |
| Aktor | | Pengguna (Programmer) | |
| Deskripsi | | Memperbarui aplikasi yang sudah ada dengan menggunakan framework | |
| Pre-condition | | Pengguna ingin memperbarui aplikasi yang sudah ada dengan menggunakan framework | |
| Post-condition | | Pengguna telah memperbarui aplikasi yang sudah ada dengan menggunakan framework | |
| **Skenario Normal** | | | |
| No. | Aksi Aktor | | Reaksi Sistem |
| 1. | Pengguna memperbarui aplikasi yang sudah ada dengan menggunakan framework | |  |
| 2. |  | | Membaca framework hasil pemasangan terhadap aplikasi yang lama |

1. Aplikasi

Berikut use case diagram beserta skenario *use case* dari aplikasi:



Gambar 3–5. Use Case Diagram Pengguna Akhir

Skenario Use Case dari Aplikasi:

Tabel 3‑4. Tabel Skenario Use Case Aplikasi 1

|  |  |  |  |
| --- | --- | --- | --- |
| No. | | 1 | |
| Nama Use Case | | Melakukan Body Checkup | |
| Aktor | | Pengguna Akhir | |
| Deskripsi | | Melakukan Body Checkup untuk mengetahui kondisi pengguna | |
| Pre-condition | | Aktor ingin melakukan body checkup | |
| Post-condition | | Aktor telah melakukan body checkup | |
| **Skenario Normal** | | | |
| No. | Aksi Aktor | | Reaksi Sistem |
| 1. | Pengguna melakukan Body checkup dengan mengisi data | |  |
| 2. |  | | Sistem akan merespon dengan cara mencocokan data pengguna dengan database sistem |
| 3. | Pengguna menerima hasil dari body checkup | |  |
| 4. |  | | Sistem memberikan hasil dari body checkup |

Tabel 3‑5. Tabel Skenario Use Case Aplikasi 2

|  |  |  |  |
| --- | --- | --- | --- |
| No. | | 2 | |
| Nama Use Case | | Memeriksa Keluhan Gizi | |
| Aktor | | Pengguna Akhir | |
| Deskripsi | | Memeriksa keluhan gizi dari hasil output body checkup | |
| Pre-condition | | Pengguna ingin memeriksa keluhan gizi | |
| Post-condition | | Pengguna telah memeriksa keluhan gizi | |
| **Skenario Normal** | | | |
| No. | Aksi Aktor | | Reaksi Sistem |
| 1. | Pengguna memeriksa keluhan gizi dari hasil output body checkup | |  |
| 2. |  | | Sistem mengolah hasil body checkup |
| 3. |  | | Sistem menampilkan hasil pilihan penangan yang tersedia |
| 4. | Pengguna memilih solusi yang akan di gunakan | |  |
| 5. |  | | Sistem mengolah pilhan pengguna |

Tabel 3‑6. Tabel Skenario Use Case Aplikasi 3

|  |  |  |  |
| --- | --- | --- | --- |
| No. | | 3 | |
| Nama Use Case | | Memilih penanganan yang tepat sesuai keluhan | |
| Aktor | | Pengguna Akhir | |
| Deskripsi | | Memilih penangan yang tepat sesuai keluhan berdasarkan hasil pemeriksaan keluhan gizi | |
| Pre-condition | | Pengguna ingin mendapatkan penangan yang tepat sesuai keluhan | |
| Post-condition | | Pengguna telah mendapatkan penangan yang tepat sesuai keluhan | |
| **Skenario Normal** | | | |
| No. | Aksi Aktor | | Reaksi Sistem |
| 1. | Pengguna memilih penangan yang tepat sesuai keluhan berdasarkan hasil pemeriksaan keluhan gizi | |  |
| 2. |  | | Sistem mengolah pilihan pengguna |
| 3. |  | | Sistem menampilkan hasil pilihan pengguna |

# **DAFTAR PUSTAKA**

|  |  |
| --- | --- |
| [1] | R. A. Supono, Karmilasari dan Y. D. Wulandari, “Aplikasi Penghitungan Kebutuhan Gizi Lansia Berbasis Smartphone Android,” *Seminar Nasional Aplikasi Teknologi Informasi (SNATi),* vol. I, p. 17, 2015. |
| [2] | I. Marlena dan E. Suryano, Ilmu Gizi, Jakarta Selatan, Jakarta: Kementrian Kesehatan Republik Indonesia, 2016, p. 182. |
| [3] | Purnomo, Sudjino, Trijoko and S. Hadisusanto, Biologi Kelas XI Untuk SMA dan MA, vol. VII, Jakarta, DKI Jakarta: Pusat Perbukuan Departemen Nasional, 2009, pp. 194-200. |
| [4] | Misnawati, "Aplikasi Penyedia Informasi Kebutuhan Gizi Orang Dewasa Berbasis Android," 2013. |
| [5] | D. Novianty dan D. Prastya, “Suara.com,” 20 Mei 2021. [Online]. Available: https://www.suara.com/tekno/2021/05/20/061609/pengguna-android-di-dunia-tembus-3-miliar?page=all. [Diakses 17 Juni 2021]. |
| [6] | A. T. Sondha, U. Sa’adah, F. F. Hardiansyah and M. B. A. Rasyid, "Framework and Code Generator for Android Development with Clean Architecture Principles Implementation," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi,* 2020. |
| [7] | GlobalStats, "statcounter," 17 June 2021. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/indonesia. [Accessed 17 June 2021]. |
| [8] | S. Vojislav, M. Milic and S. Vlajić, "Guidelines for Framework Development Process," *Conference Paper,* 2011. |
| [9] | B. A. Santoso, "medium.com," 18 October 2019. [Online]. Available: https://lobothijau.medium.com/arsitektur-mvc-vs-mvp-vs-mvvm-di-pemrograman-android-387d9c99e893. [Accessed 24 June 2021]. |
| [10] | ICHI.PRO, "ICHI.PRO," ICHI.PRO, [Online]. Available: https://ichi.pro/id/pola-arsitektur-android-bagian-3-model-view-viewmodel-255013388990267. [Accessed 24 June 2021]. |
| [11] | JitPack.io, "JitPack.io," [Online]. Available: https://jitpack.io/docs/. [Accessed 24 June 2021]. |
| [12] | R. Saraswati, “SehatQ,” Kesehatan, 16 Desember 2019. [Online]. Available: https://www.sehatq.com/artikel/pengertian-gizi-yang-mungkin-belum-anda-pahami. [Diakses 22 April 2021]. |
| [13] | M. R. Adani, “Sekawan Media,” Startup Digital, 7 Agustus 2020. [Online]. Available: https://www.sekawanmedia.co.id/pengertian-framework/. [Diakses 22 April 2021]. |
| [14] | M. S. Arif, A. Musthafa and D. Muriyatmoko, "Implementation of Model-View-ViewModel (MVVM) Architecture Pattern in the Sistem Informasi Akademik UNIDA Gontor Mobile Application," November 2019. |
| [15] | B. S. Panca, S. Mardiyanto and B. Hendradjaya, "Evaluation of Software Design Pattern on Mobile Application Based Service Development Related to the Value of Maintainability and Modularity". |
| [16] | ID Cloud Host, "ID Cloud Host," [Online]. Available: https://idcloudhost.com/panduan/mengenal-apa-itu-framework-codeigniter/. [Accessed 24 June 2021]. |
| [17] | Developer Android Google, "Developer Android Google," [Online]. Available: https://developer.android.com/topic/libraries/architecture?hl=id. [Accessed 24 June 2021]. |
| [18] | Code Tutsplus, "Code Tutsplus," [Online]. Available: https://code.tutsplus.com/id/tutorials/introduction-to-android-architecture--cms-28749. [Accessed 24 June 2021]. |
| [19] | Binus University School Of Information System, "Binus University School Of Information System," 17 March 2020. [Online]. Available: https://sis.binus.ac.id/2020/03/17/design-thinking-pengertian-tahapan-dan-contoh-penerapannya/. [Accessed 24 June 2021]. |
| [20] | K. Sokolova, M. Lemercier and L. Garcia, "Towards High Quality Mobile Applications: Android Passive MVC Architecture," *International Journal on Advances in Software,* vol. 7, no. 1-2, pp. 123-138, 2014. |
| [21] | F. E. Sahbudin and F.-F. Chua, "Design Patterns for Developing High Efficiency Mobile Application," *Jurnal of Information Technology & Software Engineering,* vol. 3, no. 3, pp. 1-9, 2013. |
| [22] | A. Daodi, G. ElBoussaidi, N. Moha and S. Kpodjedo, "An Exploratory Study of MVC-based Architectural Patterns in Android Apps," 2019. |
| [23] | M. R. Wick and A. T. Phillips, "Comparing the Template Method and Strategy Design Patterns in a Genetic Algorithm Application," *ACM SIGCSE Bulletin,* Descember 2002. |

# **LAMPIRAN**

Link Project : <https://github.com/amirisback/general-framework>

Link Jurnal : <https://github.com/amirisback/general-framework/tree/master/docs/journal>

Link Progress : <https://github.com/amirisback/general-framework/commits/master>