



KERJA PRAKTIK - EF234603

HELISA : Aplikasi Pemantauan Kesehatan Untuk Pasien

Fakultas Keperawatan Universitas Airlangga Kampus C Merr Jl Ir.
Soekarno, Mulyorejo Kota Surabaya, Jawa Timur 60115
Periode: 11 Juli 2024 - 30 Oktober 2024

Oleh:

Mochammad Taslam Gustino Prianto

5025211011

Pembimbing Departemen

Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D.

Pembimbing Lapangan

Dr. Shoffi Izza Sabilla, S.Kom.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2024



KERJA PRAKTIK - EF234603

HELISA : Aplikasi Pemantauan Kesehatan Untuk Pasien

Fakultas Keperawatan Universitas Airlangga Kampus C Merr Jl Ir.
Soekarno, Mulyorejo Kota Surabaya, Jawa Timur 60115
Periode: 11 Juli 2024 - 30 Oktober 2024

Oleh:

Mochammad Taslam Gustino Prianto

5025211011

Pembimbing Departemen

Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D.

Pembimbing Lapangan

Dr. Shoffi Izza Sabilla, S.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI.....	iv
DAFTAR GAMBAR	viii
LEMBAR PENGESAHAN.....	xii
ABSTRAK	xiv
KATA PENGANTAR.....	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan	2
1.3. Manfaat	2
1.4. Rumusan Masalah.....	2
1.5. Lokasi dan Waktu Kerja Praktik.....	3
1.6. Metodologi Kerja Praktik.....	3
1.6.1. Perumusan Masalah.....	3
1.6.2. Studi Literatur.....	3
1.6.3. Analisis dan Perancangan Sistem	4
1.6.4. Implementasi Sistem.....	4
1.6.5. Pengujian dan Evaluasi.....	4
1.6.6. Kesimpulan dan Saran	5
1.7. Sistematika Laporan.....	5
1.7.1. Bab I Pendahuluan.....	5

1.7.2.	Bab II Profil Perusahaan.....	5
1.7.3.	Bab III Tinjauan Pustaka	5
1.7.4.	Bab IV Infrastruktur Sistem.....	5
1.7.5.	Bab V Implementasi Sistem	5
1.7.6.	Bab VI Pengujian dan Evaluasi	5
1.7.7.	Bab VII Kesimpulan dan Saran	5
BAB II PROFIL PERUSAHAAN		8
2.1.	Profil Fakultas Keperawatan Universitas Airlangga....	8
2.2.	Lokasi.....	8
BAB III TINJAUAN PUSTAKA.....		10
3.1.	Pemrograman Mobile.....	10
3.2.	Dart	10
3.3.	Flutter	11
3.4.	Figma	11
BAB IV INFRASTRUKTUR SISTEM		13
4.1.	Desain Sistem.....	13
4.2.	Fungsionalitas Aplikasi HELISA.....	14
BAB V IMPLEMENTASI SISTEM		17
5.1.	Implementasi Aplikasi HELISA	17
5.1.1.	Halaman Utama (<i>Home Page</i>).....	17
5.1.2.	Pemantauan Makanan (Diet dan Cairan).....	19
5.1.3.	Pemantauan Merokok	20
5.1.4.	Menu Edukasi	21

5.1.5. Sistem Poin	22
BAB VI PENGUJIAN DAN EVALUASI	25
6.1. Tujuan Pengujian	25
6.2. Kriteria Pengujian	25
6.3. Skenario Pengujian	25
6.4. Evaluasi Pengujian.....	26
BAB VII KESIMPULAN DAN SARAN	28
7.1. Kesimpulan	28
7.2. Saran	28
LAMPIRAN	31
DAFTAR PUSTAKA.....	59
BIODATA PENULIS.....	61

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 1 Desain Arsitektur Aplikasi	13
Gambar 2 Halaman Utama	18
Gambar 3 Menu Pemantauan Makanan	20
Gambar 4 Menu Kuesioner Merokok.....	21
Gambar 5 Menu Edukasi	22
Gambar 6 Sistem Poin.....	23

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 1 Hasil Evaluasi Pengujian.....	26
---------------------------------------	----

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

HELISA : Aplikasi Pemantauan Kesehatan Untuk Pasien

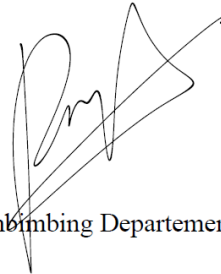
Oleh:

Mochammad Taslam Gustino Prianto

5025211011

Disetujui oleh Pembimbing Kerja Praktik:

1. Royyana Muslim Ijtihadie,
S.Kom., M.Kom., Ph.D.
NIP. 197708242003041001



(Pembimbing Departemen)

2. Dr. Shoffi Izza Sabilla,
S.Kom.
NIP. 2022199412054



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

HELISA : Aplikasi Pemantauan Kesehatan Untuk

Pasien

Nama Mahasiswa : Mochammad Taslam Gustino P.
NRP : 5025211011
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Royyana Muslim Ijtihadie,
S.Kom., M.Kom., Ph.D.
Pembimbing Lapangan : Dr. Shoffi Izza Sabilla, S.Kom.

ABSTRAK

Aplikasi HELISA dikembangkan untuk membantu pasien dalam memantau kesehatan hidup mereka setiap harinya. Aplikasi ini memungkinkan pasien untuk merekam kegiatan mereka setiap hari, seperti kesehatan tekanan darah, indeks masa tubuh, diet dan cairan, aktivitas fisik, merokok, kesehatan mental, istirahat, dan tidur. Juga di aplikasi ini pasien dapat membuat sebuah jadwal minum obat untuk mengingatkan melalui notifikasi aplikasi sesuai dengan resep obat.

Proyek ini menggunakan framework Flutter dan arsitektur Model-View-ViewModel (MVVM) untuk menangani bagian frontend. Frontend terintegrasi dengan backend melalui Application Programming Interface (API) yang dibuat oleh backend. Fitur utama aplikasi ini yaitu memantau kesehatan pasien dengan menjawab kuisisioner yang ada di dalam aplikasi.

Kata Kunci : Flutter, API, Kesehatan, Aplikasi, Kuisisioner

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: “HELISA : Aplikasi Pemantauan Kesehatan Untuk Pasien”.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Bapak Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D. selaku dosen pembimbing kerja praktik sekaligus koordinator kerja praktik.
3. Dr. Shoffi Izza Sabilla, S.Kom. selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 1 November 2024
Mochammad Taslam Gustino Prianto

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Kesehatan adalah hal yang sangat diperlukan untuk setiap orang. Sekarang orang-orang lebih peduli dengan kesehatan mereka. Dengan bertambah banyaknya makanan yang tidak sehat beredar, banyak menimbulkan penyakit.

Berobat dan konsultasi ke dokter adalah salah satu cara yang bisa dilakukan pada saat ini. Biaya berobat bisa dibilang cukup mahal. Hal tersebut juga tergantung dengan penyakit yang diderita oleh seseorang. Agar penyakit tidak kambuh lagi, maka diperlukan pemantauan oleh pasien sendiri. Pemantauan kesehatan terdiri dari tekanan darah, indeks masa tubuh, makanan sehari-hari, aktivitas fisik, konsumsi rokok, kesehatan mental, istirahat, dan lain sebagainya.

Supaya pemantauan secara mandiri bisa berjalan dengan teratur, maka dibuatlah sebuah aplikasi untuk pemantauan kesehatan. Aplikasi ini dibuat bagi pasien yang ingin memantau kesehatan mereka setiap hari. Pasien harus mengisi dengan inisiatif sendiri demi pemantauan secara maksimal. Diharapkan dengan adanya aplikasi ini pasien lebih mudah untuk memantau kesehatannya. Dari hasil pemantauan juga bisa ditunjukkan kepada dokter apabila terkena penyakit. Dokter akan menganalisis kesehatan dari hasil pemantauan yang ada di aplikasi.

1.2. Tujuan

Tujuan kerja praktik ini adalah menyelesaikan kewajiban nilai kerja praktik sebesar 4 sks dan membantu salah satu tim *research* Fakultas Keperawatan Universitas Airlangga untuk membuat aplikasi dari idenya.

1.3. Manfaat

Manfaat yang diperoleh dari kerja praktik di Fakultas Keperawatan Universitas Airlangga antara lain:

1. Meningkatkan pemahaman cara kerja dari suatu sistem dan menganalisis kebutuhan pengguna dalam pengembangan suatu aplikasi.
2. Memberikan pengalaman kerja untuk penulis dalam pengembangan perangkat lunak dalam suatu perusahaan serta memberikan pengalaman untuk developer presentasi yang baik.
3. Menambah pengalaman kerja penulis dalam pengembangan perangkat lunak.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana cara mengidentifikasi kebutuhan pengguna dari tim Fakultas Keperawatan Universitas Airlangga dan mengimplementasikannya ke dalam sistem yang dirancang agar aplikasi dapat berfungsi sesuai dengan kebutuhan pengguna?

2. Bagaimana cara merancang aplikasi HELISA agar dapat mempermudah pasien untuk memantau kesehatan?

1.5. Lokasi dan Waktu Kerja Praktik

Kegiatan kerja praktik dilaksanakan secara *hybrid* (*offline* dan *online*) di gedung Fakultas Kedokteran dan Kesehatan, di ITS Keputih, Sukolilo, Surabaya, Jawa Timur. Adapun kerja praktik dilaksanakan pada periode Juli sampai Oktober 2024. Karena pelaksanaan kerja praktik dilakukan bersamaan dengan perkuliahan semester Ganjil 2024/2025 dan mayoritas dilakukan secara *work from home*, waktu pelaksanaan tidak diatur secara ketat.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi :

1.6.1. Perumusan Masalah

Untuk mengetahui kebutuhan dari aplikasi, penulis dan tim mendiskusikan dokumen yang diberikan dari ide yang sudah ada bersama *client*, tim Fakultas Keperawatan Universitas Airlangga. Dari dokumen kebutuhan tersebut, dijelaskan konsep dan proses bagaimana pemantauan kesehatan yang diinginkan sesuai dengan ide dari *client*.

1.6.2. Studi Literatur

Setelah mendapat gambaran bagaimana sistem tersebut berjalan, penulis dan tim menawarkan tinjauan

apa saja yang akan diimplementasikan untuk membuat aplikasi beroperasi. Tinjauan yang dipakai meliputi Flutter dengan DBMS Postgres menggunakan layanan Google Cloud. Selain itu, penulis dan tim dijelaskan aturan-aturan dalam menuliskan konfigurasi agar konfigurasi dapat mudah dipahami oleh pengembang yang lain.

1.6.3. Analisis dan Perancangan Sistem

Setelah tinjauan yang dipakai telah diberitahu, untuk merancang sistem yang baik perlu adanya sebuah desain arsitektur sistem. Pada website ini tim developer setuju menggunakan arsitektur desain MVVM (Model - View - ViewModel).

1.6.4. Implementasi Sistem

Implementasi merupakan realisasi dari tahap perancangan. Pada tahap ini penulis mengimplementasikan desain UI ke dalam aplikasi Android, termasuk menerapkan kebutuhan *client*.

1.6.5. Pengujian dan Evaluasi

Setelah aplikasi yang telah direncanakan telah jadi, perlu adanya evaluasi untuk menguji apakah aplikasi sesuai dengan harapan *client*. Jika masih belum sesuai atau perlu menambah fitur, rapat akan dilakukan lagi untuk evaluasi berikutnya. Evaluasi dilakukan dengan cara melakukan *meeting* dengan tim Fakultas Keperawatan Universitas Airlangga dan mempresentasikan progres dari aplikasi yang dibuat.

1.6.6. Kesimpulan dan Saran

Pengujian yang dilakukan ini telah memenuhi syarat yang diinginkan, dan berjalan dengan baik dan lancar.

1.7. Sistematika Laporan

1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Perusahaan

Bab ini berisi gambaran umum mengenai tim yang memiliki ide sistem dari Fakultas Keperawatan Universitas Airlangga.

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Infrastruktur Sistem

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.

1.7.5. Bab V Implementasi Sistem

Bab ini berisi penjelasan tahap-tahap yang dilakukan untuk proses implementasi aplikasi.

1.7.6. Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil Fakultas Keperawatan Universitas Airlangga

Fakultas Keperawatan Universitas Airlangga (FKp UNAIR) berdiri sejak 2008. Fakultas ini mendidik mahasiswa untuk siap menjadi perawat profesional, baik secara keilmuan, sikap, dan keterampilan. FKp UNAIR berkolaborasi dengan Faculty of Medicine University Malaya. Kolaborasi yang dilakukan yaitu “Innovation in Caring Project”. Dalam kolaborasi ini FKp UNAIR membantu pasien untuk bisa memantau kesehatan mereka secara mandiri.

2.2. Lokasi

Fakultas Keperawatan Universitas Airlangga Kampus C Merr Jl Ir. Soekarno, Mulyorejo Kota Surabaya, Jawa Timur 60115.

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. Pemrograman Mobile

Mobile programming sangat berkembang seiring penggunaan *smartphone* oleh semua kalangan. Mobile programming adalah proses pengembangan aplikasi yang khususkan untuk perangkat bergerak seperti Android dan iOS.

Dalam mengembangkan aplikasi mobile, banyak teknologi yang bisa digunakan, contohnya Flutter, Kotlin, Swift, dan lain sebagainya. Bahasa pemrograman dalam setiap teknologi tersebut berbeda-beda. Teknologi yang dipilih dalam mengembangkan aplikasi mobile harus tepat sesuai dengan fungsionalitasnya. Dengan pemahaman yang baik, developer dapat menciptakan aplikasi mobile yang efisien (MDN, 2024).

3.2. Dart

Dart adalah bahasa pemrograman yang dikembangkan oleh Google. Dart dirancang untuk pembangunan aplikasi, seperti android, iOS, web, dan desktop. Untuk mengembangkan aplikasi pada Android, maka digunakan Dart sebagai bahasa pemrograman. Dart memiliki sintaks yang sederhana dan mendukung pemrograman berorientasi objek, menjadikannya pilihan yang efektif untuk membangun aplikasi dengan kinerja tinggi (Dart, 2024).

3.3. Flutter

Flutter adalah open source framework untuk membangun aplikasi multi-platform. Flutter dapat digunakan untuk membangun aplikasi berbasis mobile, web, desktop, bahkan sistem embedded. Framework Flutter ini memungkinkan bagi pengembang untuk membuat antarmuka yang responsif dan dinamis sesuai dengan kebutuhan. Flutter juga memiliki kemampuan rendering native dan akses langsung ke API. Hal itu memungkinkan pengembang untuk menciptakan aplikasi yang interaktif (Flutter, 2024).

3.4. Figma

Figma adalah alat desain antarmuka pengguna berbasis cloud yang saat ini sedang populer digunakan oleh UI/UX desainer. Figma memungkinkan kolaborasi secara langsung yang dapat memudahkan tim untuk membuat desain aplikasi secara bersamaan dalam satu halaman. Dalam Figma terdapat fitur yang mendukung dalam pembuatan prototipe, *wireframe*, dan desain antarmuka yang interaktif. Figma mempermudah bagi pengembang dalam menerapkan desain menjadi sebuah aplikasi (Figma, 2024).

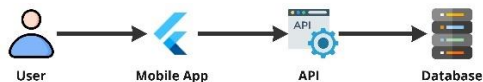
[Halaman ini sengaja dikosongkan]

BAB IV INFRASTRUKTUR SISTEM

4.1. Desain Sistem

Desain arsitektur pada aplikasi HELISA ini terbagi menjadi beberapa komponen. Aplikasi mobile dikembangkan dengan menggunakan Flutter. Flutter dipilih karena kemampuannya yang memungkinkan pembuatan antarmuka pengguna yang interaktif dan responsif. Flutter juga didukung dengan Software Development Kit (SDK) yang dapat membantu pengembang membuat aplikasi mobile. Fitur hot-reload pada Flutter memudahkan pengembang untuk memuat ulang halaman jika ada perubahan kode secara langsung. Hal tersebut dapat mempercepat dalam proses pengembangan.

Aplikasi mobile ini terintegrasi dengan API. API berguna untuk menghubungkan aplikasi mobile HELISA dengan *database*. Semua data dari pengguna akan disimpan kedalam *database*. Data yang ada di *database* juga dapat ditampilkan di dalam aplikasi melalui API tersebut. API yang telah dibuat di-*deploy* di *service* Google Cloud. Arsitektur ini memastikan aplikasi HELISA dapat menangani berbagai fungsionalitas dengan lancar.



Gambar 1 Desain Arsitektur Aplikasi

4.2. Fungsionalitas Aplikasi HELISA

Aplikasi HELISA dirancang untuk membantu pasien dalam memantau kesehatan mereka sendiri. Jika pasien mengalami sakit, ia dapat melihat apa yang dapat menyebabkan munculnya penyakit yang diderita. Aplikasi ini memiliki beberapa fitur utama yang memungkinkan pengguna untuk mencatat rekam jejak dari tekanan darah, indeks masa tubuh, makanan, aktivitas, merokok, kesehatan mental, dan istirahat setiap harinya.

Untuk mengakses aplikasi pengguna harus memiliki akun yang sudah terdaftar. Jika pengguna mengalami masalah dalam mengakses akun karena lupa kata sandi, mereka dapat menggunakan fitur “lupa password”. Fitur ini akan meminta OTP yang dikirim lewat email pengguna sebagai syarat agar dapat mengganti kata sandi yang sebelumnya.

Pada fitur yang mendukung pemantauan aktivitas, terdapat beberapa kuesioner yang harus diisi oleh pengguna secara teratur. Tidak semua pemantauan aktivitas berisi kuesioner, ada yang hanya memasukkan angka, seperti pemantauan indeks masa tubuh dan tekanan darah. Untuk pemantauan makanan, pasien tidak mengisi kuesioner seperti pemantauan lainnya. Pemantauan makanan mengisi nama makanan dan minuman yang pengguna konsumsi pada saat pagi, siang, dan malam hari. Makanan dan minuman tersebut memiliki jumlah kalori dan cairan masing-masing. Total keseluruhan kalori akan ditampilkan menggunakan indeks bar dengan maksimal kalori dan cairan harian pengguna.

Terdapat fitur utama lain untuk mengingatkan pengguna minum obat. Pengguna akan menuliskan nama obat, periode minum, dan jam minum obat sesuai dengan yang diberikan oleh resep dokter. Fitur ini memunculkan notifikasi berupa alarm sesuai jam yang dimasukkan.

Aplikasi HELISA memiliki fitur sampingan yang mendukung fitur utamanya, seperti kalender yang menunjukkan tanggal hari ini, edukasi bagi pasien tentang hidup sehat, laporan bulanan berupa grafik dari rekap kuesioner setiap harinya, edit profil pengguna, dan sistem poin yang diperoleh setelah mengisi kuesioner.

[Halaman ini sengaja dikosongkan]

BAB V

IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari sistem yang penulis buat. Hal tersebut mencakup pengembangan pada aplikasi HELISA menggunakan Flutter.

5.1. Implementasi Aplikasi HELISA

Implementasi aplikasi HELISA yang penulis kerjakan terbagi menjadi beberapa halaman, yaitu halaman utama (*home page*), pemantauan makanan (diet dan cairan), pemantauan merokok, menu edukasi, dan sistem poin.

5.1.1. Halaman Utama (*Home Page*)

Pada halaman utama ini terdapat beberapa komponen, yaitu kalender, daftar menu pemantauan, konsumsi kalori hari ini, dan daftar untuk pemantauan menu makan.

Kalender akan menampilkan semua tanggal bulan ini, beberapa tanggal di bulan sebelumnya, dan beberapa tanggal di bulan setelahnya. Terdapat dua tanda yang bisa dilihat di kalender, yaitu tanda untuk menandakan hari ini dan tanda untuk menandakan minimal mengisi satu kuesioner dari delapan menu pemantauan. Tanda untuk hari ini berbentuk bulatan pada tanggal, sedangkan tanda untuk kuesioner berbentuk titik di bawah tanggal.

Daftar menu pemantauan yang ditampilkan di halaman utama hanya empat menu saja. Menu lainnya

akan muncul di halaman lain ketika menekan tulisan “lihat semua” di pojok menu pemantauan pada halaman utama. Masing-masing tombol tersebut jika ditekan akan beralih ke halaman kuesioner yang sesuai. Konsumsi kalori hari ini ditampilkan dalam bentuk indeks bar. Persentase indeks bar yang ditampilkan sesuai dengan kebutuhan konsumsi kalori harian dan konsumsi kalori hari ini. Juga terdapat daftar *shortcut* untuk bisa langsung mengisi menu makan pagi, siang, dan malam hari.

Hi, ino



Gambar 2 Halaman Utama

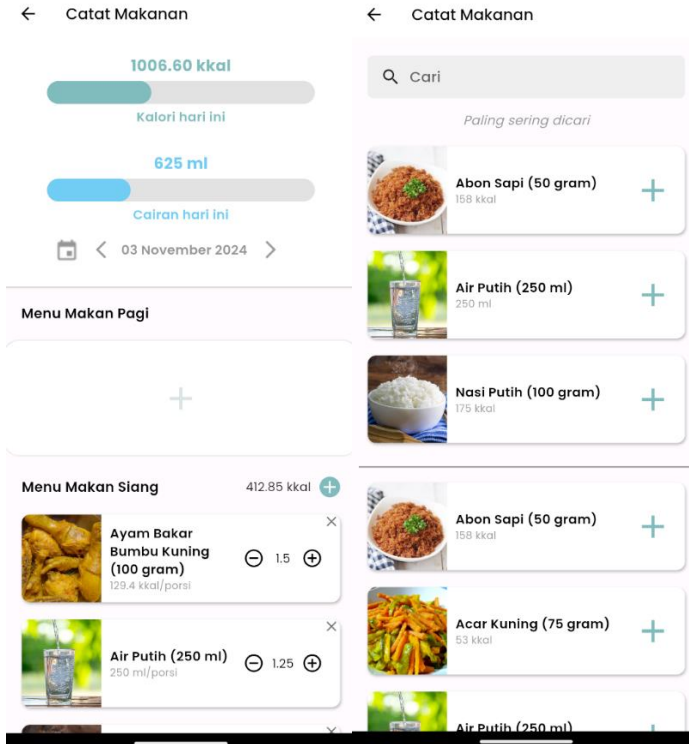
5.1.2. **Pemantauan Makanan (Diet dan Cairan)**

Pada halaman pemantauan makanan ini terdapat beberapa komponen, yaitu kalori hari ini, cairan hari ini, tanggal hari ini, menu makan hari ini (pagi, siang, dan sore).

Komponen kalori dan cairan hari ini ditampilkan dengan menggunakan indeks bar. Presentasi dari indeks bar yang ditampilkan sesuai dengan kebutuhan konsumsi kalori/cairan harian dan konsumsi kalori/cairan hari ini.

Komponen tanggal hari ini tidak hanya menampilkan tanggal saja, tetapi terdapat fitur untuk menampilkan pemantauan makanan pada tanggal sebelumnya. Tanda panah di dekat tanggal dapat ditekan untuk menampilkan pemantauan makanan tanggal sebelumnya.

Menu makan hari ini menampilkan daftar makanan yang sudah ditambahkan oleh pengguna. Makanan yang sudah ditambahkan dapat diatur porsi nya sesuai dengan yang dikonsumsi oleh pengguna. Jika ingin menambahkan makanan, bisa ditekan tombol “+”. Setelah menekan tombol “+” untuk menambahkan makanan akan dialihkan ke halaman yang memunculkan semua menu makanan yang tersedia di *database*. Terdapat fitur untuk mencari makanan dengan mengetik di *field* “Cari”. Dalam menampilkan semua menu makanan, digunakan teknik *pagination*. Teknik tersebut dapat mengurangi beban pada *device* pengguna agar tidak menampilkan semua menu sekaligus.



Gambar 3 Menu Pemantauan Makanan

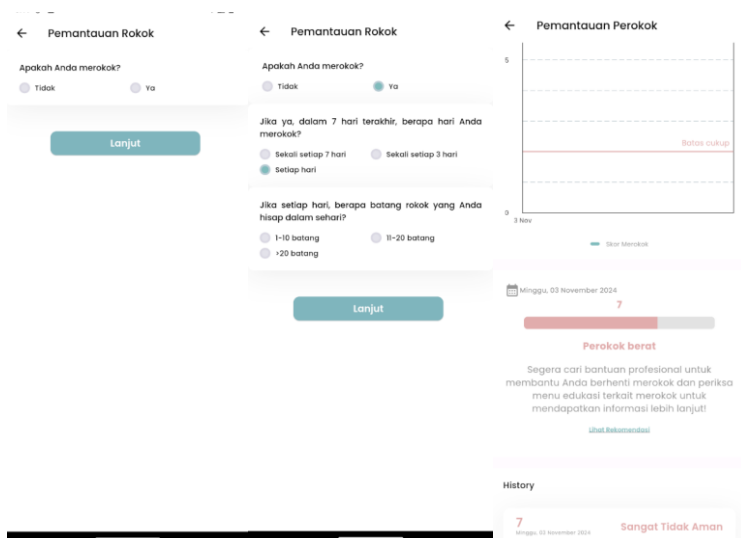
5.1.3. Pemantauan Merokok

Pada halaman pemantauan merokok terdapat dua komponen yang berbeda halaman. Komponen tersebut terdiri dari komponen kuesioner dan komponen hasil dari menjawab kuesioner.

Komponen kuesioner ini terdiri dari tiga pertanyaan yang dirancang untuk memudahkan pengguna dalam mengisi kuesioner tentang merokok. Pertanyaan akan ditampilkan secara bertahap, di mana pertanyaan berikutnya hanya akan muncul jika

pengguna memberikan jawaban tertentu pada pertanyaan sebelumnya.

Komponen hasil kuesioner akan memunculkan grafik, skor, saran, dan histori. Grafik yang ditampilkan adalah skor kuesioner yang sudah dijawab dalam bulan ini. Skor ditampilkan dengan menggunakan angka yang sesuai dengan penilaian kuesioner dan indeks bar dengan persentase yang sesuai. Saran ditampilkan sesuai dengan skor yang diperoleh pengguna. Histori akan menampilkan skor dan tanggal selama satu bulan terakhir yang telah diisi oleh pengguna.



Gambar 4 Menu Kuesioner Merokok

5.1.4. Menu Edukasi

Halaman menu edukasi berisi informasi mengenai penyakit Hipertensi dan Gaya Hidup Sehat. Pada menu ini pengguna dapat mempelajari tentang penyakit

Hipertensi, seperti definisi, faktor, gejala, diagnosis, dan lain sebagainya. Halaman ini bersifat statis, semua informasi ditangani pada *frontend*.



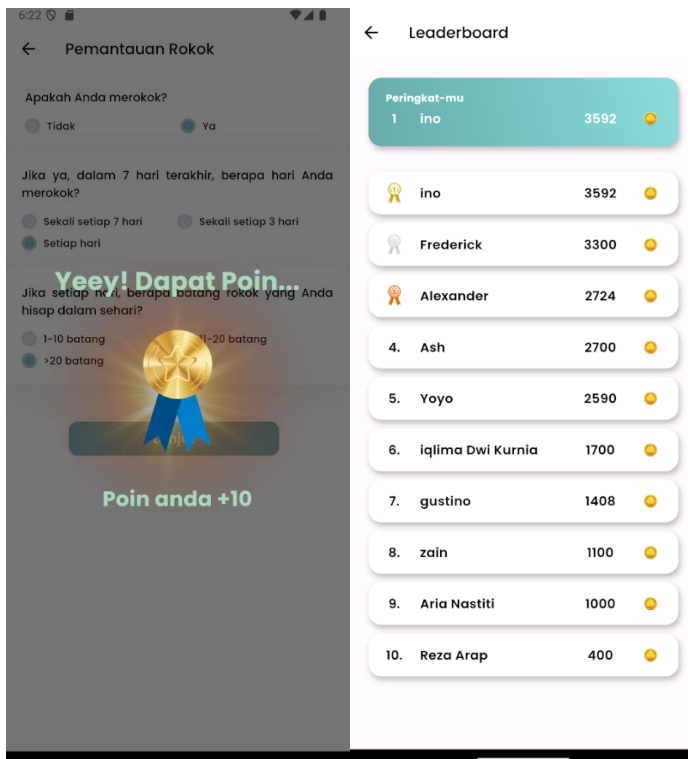
Gambar 5 Menu Edukasi

5.1.5. Sistem Poin

Pada sistem poin ini pengguna akan mendapat poin jika selesai mengisi kuesioner dan meminum obat sesuai dengan jadwal yang dijadwalkan. Poin tidak hanya bisa bertambah, tetapi bisa berkurang. Hal tersebut bisa terjadi pada menu pemantauan makanan.

Pengurangan dan penambahan poin diatur langsung oleh *backend*.

Terdapat halaman untuk menampilkan total poin yang didapat oleh pengguna dan *leaderboard* pengguna. *Leaderboard* tersebut menampilkan peringkat yang didapat oleh pengguna itu sendiri dan pengguna dengan peringkat sepuluh teratas.



Gambar 6 Sistem Poin

[Halaman ini sengaja dikosongkan]

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap Aplikasi HELISA. Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi arsitektur dengan analisis dan perancangan arsitektur.

6.1. Tujuan Pengujian

Pengujian dilakukan terhadap Aplikasi HELISA guna menguji kemampuan arsitektur dalam melayani permintaan sistem aplikasi.

6.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut :

- a. Kemampuan aplikasi untuk menampilkan antar muka pengguna.
- b. Kemampuan aplikasi untuk menampilkan data yang diambil dari *database*.
- c. Kemampuan aplikasi untuk menyimpan data ke *database*.
- d. Kemampuan aplikasi agar bisa efektif dalam menampilkan data yang banyak.

6.3. Skenario Pengujian

Skenario pengujian dilakukan dengan melakukan peran sebagai user yang akan menjalankan fitur-fitur.

Langkah-langkah untuk setiap kebutuhan fungsionalitas yaitu sebagai berikut :

1. Pengguna dapat membuka aplikasi HELISA
2. Pengguna dapat login ke aplikasi HELISA
3. Pengguna dapat mengisi kuesioner setiap menu
4. Pengguna dapat melihat hasil dari pengisian kuesioner

6.4. Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sistem aplikasi PPDB terhadap kasus skenario uji coba. Tabel 6.1 di bawah ini menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.

Tabel 1 Hasil Evaluasi Pengujian

Kriteria Pengujian	Hasil Pengujian
Aplikasi dapat melayani antar muka pengguna	Terpenuhi
Aplikasi dapat menampilkan data yang diambil dari database.	Terpenuhi
Aplikasi dapat menyimpan data ke database.	Terpenuhi
Aplikasi dapat menampilkan data yang banyak secara efektif.	Terpenuhi

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan pengembangan aplikasi HELISA adalah sebagai berikut :

- a. Melalui serangkaian wawancara dan diskusi dengan tim Fakultas Keperawatan Universitas Airlangga, penulis berhasil mengidentifikasi kebutuhan pengguna dan mengimplementasikan ke dalam sistem yang dirancang. Hal ini terbukti dari hasil pengujian yang menunjukkan bahwa aplikasi berfungsi sesuai dengan kebutuhan pengguna.
- b. Aplikasi HELISA telah memudahkan pasien dalam memantau kesehatan mereka. Sehingga pasien dapat secara efisien menggunakan aplikasi HELISA. Fitur utama pada aplikasi ini yaitu mencatat rekam jejak dari tekanan darah, indeks masa tubuh, makanan, aktivitas, merokok, kesehatan mental, dan istirahat setiap harinya. Dengan desain antarmuka yang cukup simpel dan informasi yang jelas, pasien akan terbantu untuk memantau kesehatan mereka.

7.2. Saran

Saran untuk aplikasi HELISA adalah sebagai berikut :

- a. Optimalisasi antarmuka pengguna untuk memberikan pengalaman yang lebih baik.

- b. Mengembangkan sistem *reward* untuk pengguna agar lebih memotivasi untuk memantau kesehatan.
- c. Menambahkan fitur lain yang dapat membantu pengguna dalam memantau kesehatan secara mandiri.
- d. Menuliskan dokumentasi kebutuhan pengguna untuk memudahkan pemeliharaan.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

1. Bukti Penerimaan Kerja Praktik



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS AIRLANGGA
FAKULTAS KEPERAWATAN
Kampus C Mulyorejo Surabaya 60115 Telp. 031-5913756, Fax. 031-5913752
Laman : <http://nrs.unair.ac.id> email : huwas@fkip.unair.ac.id

SURAT KETERANGAN
Nomor : 7997/TB/UN3.FKp/PK.01.06/2024

Yang bertanda tangan di bawah ini:

Nama : Prof. Dr. Ah. Yusuf S., S.Kp., M.Kes
NIP : 196701012000031002
Golongan/Pangkat : Pembina Utama Madya (IV/d)
Jabatan : Dekan Fakultas Keperawatan Universitas Airlangga

Menerangkan dengan sebenarnya bahwa:

Nama : Mochammad Taslam Gustino Prianto
NRP : 5025211011
Program Studi : Teknik Informatika
Fakultas : Fakultas Teknologi Elektro dan Informatika Cerdas
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

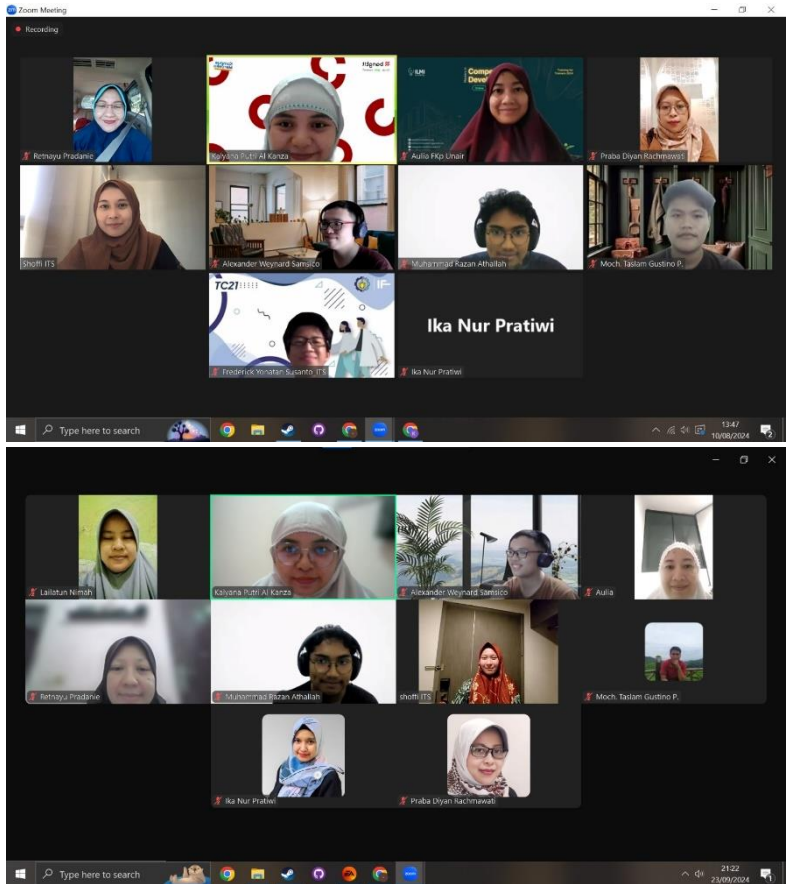
telah melaksana Kerja Praktek (KP) di Fakultas Keperawatan Universitas Airlangga pada tanggal 11 Juli 2024 sampai dengan 30 Oktober 2024.

Demikian surat keterangan ini kami buat dengan sebenar-benarnya untuk dapat digunakan sebagaimana mestinya.

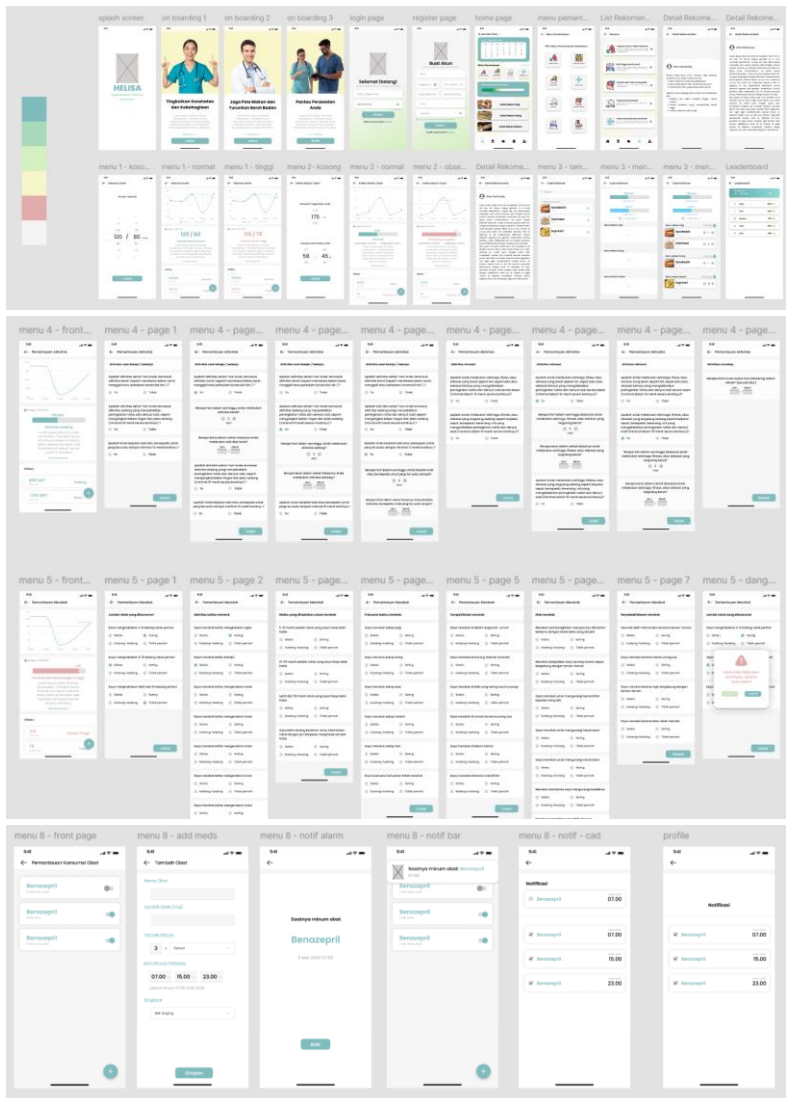
12 November 2024
Dekan

Prof. Dr. Ah. Yusuf S., S.Kp., M.Kes
NIP 196701012000031002

2. Dokumentasi Kerja Praktik



3. Desain Antarmuka Aplikasi



4. Dokumentasi Kode Sampel

- Halaman Utama

```

class HomePage extends StatefulWidget {
  HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  final ApiConfig _apiClient = ApiConfig();
  List<DateTime> _markedDays = [];

  // Table_calendar
  DateTime _today = DateTime.now();
  DateTime? _selectedDay;

  // Tanggal Hari Ini
  late String dayName;
  late String day;
  late String month;
  late String monthName;
  late String year;
  late String _apiDate;

  // Initiate fetch
  Future<void> fetchData() async {
    try {
      final String testMonth = "$year-$month";
      final respTestPerMonth = await _apiClient.getTestPerMonth(testMonth);
      final List<String> data = respTestPerMonth.data!;

      setState(() {
        _markedDays = data.map((date) {
          final parts = date.split('-');
          return DateTime.utc(
            int.parse(parts[0]),
            int.parse(parts[1]),
            int.parse(parts[2]),
          );
        }).toList();
      });
      print("MarkedDay : $_markedDays");
    } catch (e) {
      print(e);
    }
  }
}

```

```

@override
void initState() {
  // TODO: implement initState
  super.initState();
  dayName = DateFormat('EEEE', "id_ID").format(_today);
  day = DateFormat('dd').format(_today);
  month = DateFormat('MM').format(_today);
  monthName = DateFormat('MMMM', "id_ID").format(_today);
  year = DateFormat('y').format(_today);
  _apiDate = '$year-$month-$day';
  fetchData();
}

@override
Widget build(BuildContext context) {
  final double deviceWidth = MediaQuery.of(context).size.width;
  final double deviceHeight = MediaQuery.of(context).size.height;

  return Scaffold(
    appBar: AppBar(
      title: FutureBuilder<ProfileResponse>(
        future: _apiClient.getProfile(),
        builder: (context, snapshot) {
          if (snapshot.connectionState == ConnectionState.waiting) {
            return const Text(
              'Loading ...',
              style: TextStyle(fontSize: 15, color: Colors.grey),
            ); // Text
          } else if (snapshot.hasError) {
            WidgetsBinding.instance.addPostFrameCallback(() {
              Navigator.pushNamedAndRemoveUntil(
                context, "/onboarding", (r) => false;
              );
            });
            return const Center(
              child: Text(
                'Error: Something went wrong',
                style: TextStyle(
                  color: Colors.red,
                  fontSize: 12.0,
                  fontWeight: FontWeight.w500), // TextStyle
            ); // Text // Center
          } else if (!snapshot.hasData || snapshot.data!.data == null) {
            return const Center(
              child: Text(
                'No data available',
                style: TextStyle(fontSize: 15, color: Colors.grey),
            ); // Text // Center
          } else {
            String nama = snapshot.data!.data!.nama!;
            return Text(
              'Hi, $nama',
              style: const TextStyle(
                fontSize: 18,

```

```

        style: const TextStyle(
          fontSize: 18,
          fontWeight: FontWeight.w600,
          color: Colors.black,
        ), // TextStyle
      ); // Text
    }
  )), // FutureBuilder
  automaticallyImplyLeading: false,
  backgroundColor: Colors.white,
  scrolledUnderElevation: 0.0,
), // AppBar
body: Stack(
  children: [
    Positioned.fill(
      child: Image.asset(
        'assets/home/screen_background.png',
        fit: BoxFit.cover,
      ), // Image.asset
    ), // Positioned.fill
    SingleChildScrollView(
      padding: EdgeInsets.fromLTRB(
        deviceWidth * 0.08, 20, deviceWidth * 0.08, 70), // EdgeInsets
      child: Column(
        children: [
          Container(
            padding: EdgeInsets.fromLTRB(
              deviceWidth * 0.01,
              deviceHeight * 0.01,
              deviceWidth * 0.01,
              deviceHeight * 0.01), // EdgeInsets.fromLTRB
            width: deviceWidth,
            height: 275,
            decoration: BoxDecoration(
              color: const Color(0xFF80BCBD),
              borderRadius: BorderRadius.circular(15),
              boxShadow: const [
                BoxShadow(
                  color: Colors.black26,
                  blurRadius: 5.0,
                  offset: Offset(2, 4),
                ) // BoxShadow
              ],
            ), // BoxDecoration
          child: Padding(
            padding: EdgeInsets.fromLTRB(
              deviceWidth * 0.065,
              deviceHeight * 0.02,
              deviceWidth * 0.065,
              deviceHeight * 0.02), // EdgeInsets.fromLTRB
            child: Column(
              mainAxisAlignment: MainAxisAlignment.spaceAround.

```

```

crossAxisAlignment: CrossAxisAlignment.start,
children: [
  Padding(
    padding:
      EdgeInsets.only(bottom: deviceHeight * 0.002),
    child: Text(
      '$dayName, $day $monthName $year',
      style: const TextStyle(
        color: Colors.white,
        fontWeight: FontWeight.w600,
        fontSize: 15,
      ), // TextStyle
    ), // Text
  ), // Padding
  Container(
    padding: const EdgeInsets.all(10),
    width: deviceWidth,
    // height: deviceHeight*0.25,
    height: 190,
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),
      color: Colors.white,
    ), // BoxDecoration
    child: TableCalendar(
      focusedDay: _today,
      firstDay: DateTime.utc(2020, 1, 1),
      lastDay: DateTime.utc(2030, 12, 31),
      selectedDayPredicate: (day) =>
        isSameDay(_selectedDay, day),
      headerVisible: false,
      rowHeight: 30,
      calendarStyle: const CalendarStyle(
        defaultTextStyle: TextStyle(fontSize: 13),
        weekendTextStyle: TextStyle(fontSize: 13),
        selectedTextStyle: TextStyle(fontSize: 13),
        todayTextStyle: TextStyle(fontSize: 13),
        outsideTextStyle: TextStyle(fontSize: 9),
        todayDecoration: BoxDecoration(
          color: Color(0xFFAAD9BB),
          shape: BoxShape.circle,
        ), // BoxDecoration // CalendarStyle
      ),
      calendarBuilders: CalendarBuilders(
        markerBuilder: (context, date, events) {
          if (_markedDays.contains(date)) {
            return Positioned(
              bottom: 1,
              child: Container(
                height: 5,
                width: 5,
                decoration: const BoxDecoration(
                  color: Colors.red,
                  shape: BoxShape.circle,
                ), // BoxDecoration
            );
          }
        },
      ),
    ),
  ),
],
);

```



```

        route: '/menu1',
      ), // CardPemantauan
      CardPemantauan(
        img: AssetImage('assets/home/food.png'),
        title: 'Diet & Cairan',
        route: '/menu3',
      ), // CardPemantauan
      CardPemantauan(
        img: AssetImage('assets/home/cigarrete.png'),
        title: 'Merokok',
        route: '/menu5',
      ), // CardPemantauan
      CardPemantauan(
        img: AssetImage('assets/home/activity.png'),
        title: 'Aktivitas Fisik',
        route: '/menu4',
      ), // CardPemantauan
    ],
  ), // Row
],
), // Column
), SizedBox(height: deviceHeight * 0.02),
Container(
  padding: EdgeInsets.fromLTRB(
    deviceWidth * 0.08,
    deviceHeight * 0.02,
    deviceWidth * 0.08,
    deviceHeight * 0.02), // EdgeInsets.fromLTRB
  width: deviceWidth,
  // height: deviceHeight*0.14,
  height: 110,
  decoration: BoxDecoration(
    color: const Color(0xFF80BCBD),
    borderRadius: BorderRadius.circular(15),
    boxShadow: const [
      BoxShadow(
        color: Colors.black26,
        blurRadius: 5.0,
        offset: Offset(2, 4),
      ) // BoxShadow
    ],
  ), // BoxDecoration
  child: Column(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    crossAxisAlignment: CrossAxisAlignment.center,
    children: [
      Text(
        "Konsumsi Kalori Hari Ini",
        style: TextStyle(
          color: Colors.white,
          fontSize: 15,
          fontWeight: FontWeight.bold,
          shadows: <Shadow>[

```



```

shadows: <Shadow>[
  Shadow(
    offset: const Offset(2.0, 2.0),
    blurRadius: 10.0,
    color: Colors.black.withOpacity(0.3),
  ), // Shadow
], // <Shadow>[]
), // TextStyle
), // Text
FutureBuilder(
  future: _apiClient.getKonsumsi(_apiDate),
  builder: (context, snapshot) {
    if (!snapshot.hasData ||
        snapshot.data!.data == null) {
      return Column(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        children: [
          Container(
            width: deviceWidth,
            height: deviceHeight * 0.025,
            decoration: BoxDecoration(
              color: Colors.white,
              borderRadius: BorderRadius.circular(25)), //
            child: ClipRRect(
              borderRadius: BorderRadius.circular(20),
              child: const ProgressBar(
                value: 0,
                gradient: LinearGradient(
                  begin: Alignment.topLeft,
                  end: Alignment.topRight,
                  colors: [
                    Color(0xFF00A35E),
                    Color(0xFFE6E939),
                    Color(0xFFFF0000)
                  ]), // LinearGradient
              )), // ProgressBar // ClipRRect
          ), // Container
          Text(
            "0 kkal",
            style: TextStyle(
              color: Colors.white,
              fontSize: 13,
              fontWeight: FontWeight.bold,
              shadows: <Shadow>[
                Shadow(
                  offset: const Offset(2.0, 2.0),
                  blurRadius: 10.0,
                  color: Colors.black.withOpacity(0.3),
                ), // Shadow
              ], // <Shadow>[]
            ), // TextStyle
          ), // Text
        ],
      ),
    ),
  ),
),

```

```

} else {
  double totalKalori =
    snapshot.data!.data!.totalKalori!.toDouble();
  double targetKalori =
    snapshot.data!.data!.targetKalori!.toDouble();
  double barValue = totalKalori / targetKalori;
  return Column(
    mainAxisAlignment: MainAxisAlignment.spaceAround,
    children: [
      Container(
        width: deviceWidth,
        height: deviceHeight * 0.025,
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.circular(25)), //
        child: ClipRRect(
          borderRadius: BorderRadius.circular(20),
          child: ProgressBar(
            value: barValue,
            gradient: const LinearGradient(
              begin: Alignment.topLeft,
              end: Alignment.topRight,
              colors: [
                Color(0xFF00A35E),
                Color(0xFFE6E939),
                Color(0xFFFF0000)
              ], // LinearGradient
            )), // ProgressBar // ClipRRect
        ), // Container
      Text(
        '${totalKalori.toStringAsFixed(2)} kkal',
        style: TextStyle(
          color: Colors.white,
          fontSize: 13,
          fontWeight: FontWeight.bold,
          shadows: <Shadow>[
            Shadow(
              offset: const Offset(2.0, 2.0),
              blurRadius: 10.0,
              color: Colors.black.withOpacity(0.3),
            ), // Shadow
          ], // <Shadow>[]
        ), // TextStyle
      ), // Text
    ],
  ); // Column
}
},

```

```

    ], // Container
    SizedBox(height: deviceHeight * 0.02),
    const Column(
      children: [
        CatatMakan(
          img: AssetImage('assets/home/makan_pagi.png'),
          title: 'Catat Makan Pagi',
          waktuKonsumsi: 'Sarapan',
        ), // CatatMakan
        CatatMakan(
          img: AssetImage('assets/home/makan_siang.png'),
          title: 'Catat Makan Siang',
          waktuKonsumsi: 'Makan Siang',
        ), // CatatMakan
        CatatMakan(
          img: AssetImage('assets/home/makan_malam.png'),
          title: 'Catat Makan Malam',
          waktuKonsumsi: 'Makan Malam',
        ) // CatatMakan
      ],
    ), // Column
  ],
), // Column
), // SingleChildScrollView
// Navbar
const Positioned(
  left: 0,
  right: 0,
  bottom: 0,
  child: CustomNavbar(
    homeIsActive: true,
    progresIsActive: false,
    educationIsActive: false,
    resumeIsActive: false,
    akunIsActive: false)), // CustomNavbar // Positioned
],

```

- **Pemantauan Makanan**

```

class Menu3Page extends StatefulWidget {
  const Menu3Page({super.key});

  @override
  State<Menu3Page> createState() => _Menu3PageState();
}

class _Menu3PageState extends State<Menu3Page> {
  final ApiConfig _apiClient = ApiConfig();
  late Future<GetDataMenu3Response> _futureGetKonsumsi;
  late bool hasRefresh;

  // Tanggal hari ini
  late final DateTime _today = DateTime.now();
  late DateTime _swapDay;
  late String day;
  late String month;
  late String monthName;
  late String year;
  late String _apiDate;

  Future<void> _loadData() async {
    setState(() {
      _futureGetKonsumsi = _apiClient.getKonsumsi(_apiDate);
    });
  }

  void _refreshData() {
    setState(() {
      hasRefresh = true;
    });
  }

  void _swapDayAction() {
    setState(() {
      day = DateFormat('dd').format(_swapDay);
      month = DateFormat('MM').format(_swapDay);
      monthName = DateFormat('MMMM', "id_ID").format(_swapDay);
      year = DateFormat('y').format(_swapDay);
      _apiDate = '$year-$month-$day';
    });
  }

  @override
  void initState() {
    // TODO: implement initState
    super.initState();
    hasRefresh = false;
    _swapDay = _today;
    _swapDayAction();
    _loadData();
  }
}

```

```

@override
Widget build(BuildContext context) {
  final double deviceWidth = MediaQuery.of(context).size.width;
  final double deviceHeight = MediaQuery.of(context).size.height;

  return Scaffold(
    appBar: appBar(context, 'Catat Makanan', "/home"),
    body: FutureBuilder<GetDataMenu3Response>(
      future: _futureGetKonsumsi,
      builder: (context, snapshot) {
        if (hasRefresh == true) {
          hasRefresh = false;
          return const Center(child: CircularProgressIndicator());
        } else if (!snapshot.hasData || snapshot.data!.data == null) {
          return _NullDataPage(deviceWidth, deviceHeight);
        } else {
          // Data Tes
          dynamic totalKalori = snapshot.data!.data!.totalKalori!;
          dynamic totalCairan = snapshot.data!.data!.totalCairan!;
          dynamic targetKalori = snapshot.data!.data!.targetKalori!;
          dynamic targetCairan = snapshot.data!.data!.targetCairan!;
          List<MenuItem> sarapan = [];
          List<MenuItem> makanSiang = [];
          List<MenuItem> makanMalam = [];
          double valueKaloriBar = totalKalori / targetKalori;
          double valueCairanBar = totalCairan / targetCairan;
          double kaloriMakanPagi = 0;
          double kaloriMakanSiang = 0;
          double kaloriMakanMalam = 0;

          if (snapshot.data!.data!.sarapan != null) { ...
          }

          if (snapshot.data!.data!.makanSiang != null) { ...
          }

          if (snapshot.data!.data!.makanMalam != null) { ...
          }

          return SingleChildScrollView(
            child: Container(
              padding: const EdgeInsets.only(bottom: 20),
              child: Column(
                children: [
                  Container(
                    width: deviceWidth,
                    height: 290,
                    padding: const EdgeInsets.fromLTRB(50, 20, 50, 10),
                    margin: const EdgeInsets.only(bottom: 20),
                    decoration: const BoxDecoration(
                      color: Colors.white,
                      boxShadow: [
                        BoxShadow(
                          color: Colors.black26,

```

```

        BoxShadow(
          color: Colors.black26,
          blurRadius: 1.0,
          offset: Offset(0, 1),
        ) // BoxShadow
      ],
    ), // BoxDecoration
    child: Column(
      children: [
        Container(
          color: Colors.white,
          child: Column(
            children: [
              Text(
                '${totalKalori.toStringAsFixed(2)} kkal',
                style: const TextStyle(
                  fontSize: 18,
                  fontWeight: FontWeight.w800,
                  color: Color(0xFF80BCBD)), // TextStyle
              ), // Text
              SizedBox(
                height: deviceHeight * 0.005,
              ), // SizedBox
              Container(
                width: deviceWidth,
                height: deviceHeight * 0.03,
                decoration: BoxDecoration(
                  color: const Color(0xFFE2E2E2),
                  borderRadius:
                    BorderRadius.circular(15)), // Box
                child: ClipRRect(
                  child: ProgressBar(
                    value: valueKaloriBar,
                    color: const Color(0xFF80BCBD),
                  ), // ProgressBar
                ), // ClipRRect // Container
              SizedBox(
                height: deviceHeight * 0.005,
              ), // SizedBox
              const Text(
                'Kalori hari ini',
                style: TextStyle(
                  fontSize: 15,
                  fontWeight: FontWeight.w600,
                  color: Color(0xFF80BCBD)), // TextStyle
              ) // Text
            ],
          ), // Column
        ), // Container
        SizedBox(
          height: deviceHeight * 0.035,

```

```

Container(
  color: Colors.white,
  child: Column(
    children: [
      Text(
        '$totalCairan ml',
        style: const TextStyle(
          fontSize: 18,
          fontWeight: FontWeight.w800,
          color: Color(0xFF71CDF4)), // TextStyle
      ), // Text
      SizedBox(
        height: deviceHeight * 0.005,
      ), // SizedBox
      Container(
        width: deviceWidth,
        height: deviceHeight * 0.03,
        decoration: BoxDecoration(
          color: const Color(0xFFE2E2E2),
          borderRadius:
            BorderRadius.circular(15)), // BoxDecorat
        child: ClipRRect(
          child: ProgressBar(
            value: valueCairanBar,
            color: const Color(0xFF71CDF4),
          ), // ProgressBar
        )), // ClipRRect // Container
      SizedBox(
        height: deviceHeight * 0.005,
      ), // SizedBox
      const Text(
        'Cairan hari ini',
        style: TextStyle(
          fontSize: 15,
          fontWeight: FontWeight.w600,
          color: Color(0xFF71CDF4)), // TextStyle
      ) // Text
    ],
  ), // Column
), // Container
SizedBox(
  height: deviceHeight * 0.02,
), // SizedBox

// TANGGAL
Container(
  width: deviceWidth * 0.7,
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Image.asset(
        'assets/icons/calendar.png',

```

```

Container(
  width: deviceWidth * 0.5,
  child: Row(
    mainAxisAlignment:
      MainAxisAlignment.spaceBetween,
    children: [
      InkWell(
        onTap: () {
          setState(() {
            _swapDay = _swapDay.subtract(
              const Duration(days: 1));
            _swapDayAction();
            _refreshData();
            _loadData();
          });
        },
        child: Container(
          width: 15,
          child: const Icon(
            Icons.arrow_back_ios,
            color: Colors.grey,
            size: 25,
          ), // Icon
        ), // Container
      ), // InkWell
      Text(
        '$day $monthName $year',
        style: const TextStyle(
          fontSize: 15,
          fontWeight: FontWeight.w600,
          color: Color(0xFF969696)), // TextStyle
      ), // Text
      InkWell(
        onTap: () {
          setState(() {
            _swapDay = _swapDay
              .add(const Duration(days: 1));
            _swapDayAction();
            _refreshData();
            _loadData();
          });
        },
        child: Container(
          width: 15,
          child: const Icon(
            Icons.arrow_forward_ios,
            color: Colors.grey,
            size: 25,
          ), // Icon
        ), // Container
      ), // InkWell
    ],
  ),
),

```



```
    ), // Container
    // MENU MAKAN PAGI
    Column(...
        children: [
            sarapan.isEmpty
            ? MenuEmpty( // MenuEmpty ...
                : MenuFill( // MenuFill ...
        ]), // Column
    // MENU MAKAN SIANG
    Column(...
        children: [
            makanSiang.isEmpty
            ? MenuEmpty( // MenuEmpty ...
                : MenuFill( // MenuFill ...
        ]), // Column
    // MENU MAKAN MALAM
    Column(...
        children: [
            makanMalam.isEmpty
            ? MenuEmpty( // MenuEmpty ...
                : MenuFill( // MenuFill ...
        ]), // Column
    ],
    ), // Column
    ), // Container
    ]; // SingleChildScrollView
}
), // FutureBuilder
); // Scaffold
}

SingleChildScrollView _NullDataPage(double deviceWidth, double deviceHeight) { ...
```

- **Pemantauan Merokok**

```

class Menu7Form extends StatefulWidget {
  const Menu7Form({super.key});

  @override
  State<Menu7Form> createState() => _Menu7FormState();
}

class _Menu7FormState extends State<Menu7Form> {
  Future<storeData.StoreDataTestMenu7Response?> _futureResponse;
  final ApiConfig _apiClient = ApiConfig();

  @override
  Widget build(BuildContext context) {
    final double screenWidth = MediaQuery.of(context).size.width;
    final double screenHeight = MediaQuery.of(context).size.height;

    List<TextEditingController> controllers = [];

    return Scaffold(
      appBar: appBarAlert(context, "Pemantauan Kesehatan Mental", "/home"),
      backgroundColor: const Color(0xFFFF6F6F),
      body: SingleChildScrollView(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          children: [
            FutureBuilder<getData.GetPertanyaanMenu7Response>(
              future: _apiClient.getQuestionsMental(),
              builder: (context, snapshot) {
                if (snapshot.connectionState == ConnectionState.waiting) {
                  return const Center(child: CircularProgressIndicator());
                } else if (snapshot.hasError) {
                  return const Center(
                    child: Text(
                      'Error: Something went wrong',
                      style: TextStyle(
                        color: Colors.red,
                        fontSize: 12.0,
                        fontWeight: FontWeight.w500,
                      ), // TextStyle
                    ), // Text
                  ); // Center
                } else if (!snapshot.hasData ||
                    snapshot.data!.data == null ||
                    snapshot.data!.data!.isEmpty) {
                  return const Center(child: Text('No data available'));
                } else {
                  return Column(
                    mainAxisAlignment: MainAxisAlignment.min,
                    children: [
                      ListView.builder(
                        physics: const NeverScrollableScrollPhysics(),
                        shrinkWrap: true,
                        itemCount: snapshot.data!.data!.length,
                        itemBuilder: (context, index) {
                          getData.Data pertanyaan = snapshot.data!.data![index];

```

```

        getData.Data pertanyaan = snapshot.data!.data![index];
        controllers.add(TextEditingController());

        Widget card;
        card = MyYesNoCard(
            controller: controllers[index],
            question: pertanyaan.deskripsi.toString(),
        ); // MyYesNoCard

        return Column(
            children: [
                const SizedBox(height: 16),
                card,
            ],
        ); // Column
    },
), // ListView.builder
const SizedBox(height: 40),
Padding(
    padding: const EdgeInsets.only(right: 8.0),
    child: Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
            ElevatedButton(
                onPressed: () {
                    bool completed = true;
                    List<MentalRequest> mentalRequests =
                        controllers
                            .asMap()
                            .entries
                            .map((entry) {
                                int index = entry.key;
                                TextEditingController controller =
                                    entry.value;
                                if (controller.text == '-1') {
                                    completed = false;
                                }
                                return MentalRequest(
                                    pertanyaanId: index + 1,
                                    jawaban: int.tryParse(controller.text),
                                ); // MentalRequest
                            }).toList();
                    if (!completed) {
                        errorDialog();
                    } else {
                        setState(() {
                            _futureResponse =
                                _apiClient.postMental(mentalRequests);
                        });
                    }
                },
            ),
        ],
    ),

```

```

        style: ButtonStyle(
          backgroundColor: MaterialStateProperty.all<Color>(const Color(0xff80BCBD)),
          foregroundColor: MaterialStateProperty.all<Color>(Colors.white),
          padding: MaterialStateProperty.all<EdgeInsets>(
            const EdgeInsets.symmetric(horizontal: 100),
          ),
          shape: MaterialStateProperty.all<OutlinedBorder>(
            RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(10),
            ), // RoundedRectangleBorder
          ),
        ), // ButtonStyle
        child: const Text(
          'Lanjut',
          style: TextStyle(
            fontWeight: FontWeight.w600,
            fontSize: 16,
          ), // TextStyle
        ), // Text
      ), // ElevatedButton
    ],
  ), // Row
), // Padding
const SizedBox(height: 40),
_futureResponse == null ? Container() : buildFutureBuilder(),

```

```

void errorDialog() {
  builder: (context) => AlertDialog(
    backgroundColor: Colors.white,
    title: const Center(
      child: Text(
        'Peringatan',
        style: TextStyle(
          fontSize: 20,
          fontWeight: FontWeight.bold,
        ), // TextStyle
      ), // Text
    ), // Center
    content: const Text('Harap isi semua pertanyaan.'),
    actions: [
      TextButton(
        onPressed: () {
          Navigator.pop(context);
        },
        child: const Text(
          'OK',
          style: TextStyle(
            fontSize: 14,
            fontWeight: FontWeight.w600,
            color: Colors.black,
          ), // TextStyle
        ), // Text
      ), // TextButton
    ],
  )); // AlertDialog
}

FutureBuilder<storeData.StoreDataTestMenu7Response> buildFutureBuilder() {
  return FutureBuilder<storeData.StoreDataTestMenu7Response>(
    future: _futureResponse,
    builder: (context, snapshot) {
      if (snapshot.connectionState == ConnectionState.done) {
        if (snapshot.hasData) {
          WidgetsBinding.instance.addPostFrameCallback((_) {
            achievementDialog(context, 10);
            Future.delayed(Duration(milliseconds: 1700), () {
              Navigator.pop(context);
              Navigator.pushReplacementNamed(context, '/menu7');
            }); // Future.delayed
          });
        } else if (snapshot.hasError) {
          return const Text(
            'Error: Pastikan semua sudah terisi / coba lagi',
            style: TextStyle(
              color: Colors.red,
              fontSize: 12.0,
              fontWeight: FontWeight.w500,
            ), // TextStyle
          ); // Text
        }
      }
    }
  );
}

```

- Menu Edukasi

```
class ListEducation extends StatelessWidget {
  const ListEducation({super.key});

  @override
  Widget build(BuildContext context) {
    final double deviceWidth = MediaQuery.of(context).size.width;
    final double deviceHeight = MediaQuery.of(context).size.height;

    return Scaffold(
      appBar: appBar(context, 'Edukasi', '/home'),
      body: Stack(
        children: [
          Positioned.fill(
            child: Image.asset(
              'assets/home/screen_background.png',
              fit: BoxFit.cover,
            ), // Image.asset
          ), // Positioned.fill
          Container(
            height: deviceHeight,
            child: SingleChildScrollView(
              padding: EdgeInsets.fromLTRB(
                deviceWidth * 0.08, 20, deviceWidth * 0.08, 70), // E
              child: const Column(
                children: [
                  CardEducation(
                    title: 'Hipertensi',
                    route: '/hipertensi',
                  ), // CardEducation
                  CardEducation(
                    title: 'Gaya Hidup Sehat',
                    route: '/gaya-hidup',
                  ), // CardEducation
                ],
              ), // Column
            ), // SingleChildScrollView
          ), // Container
          // Navbar
          const Positioned(
            left: 0,
            right: 0,
            bottom: 0,
            child: CustomNavbar(
              homeIsActive: false,
              progresIsActive: false,
              educationIsActive: true,
              resumeIsActive: false,
              akunIsActive: false)), // CustomNavbar // Positioned
        ],
      ), // Stack
    ); // Scaffold
  }
}
```

- Sistem Poin

```

void achievementDialog(context, int Poin) {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return Dialog(
        surfaceTintColor: Colors.transparent,
        backgroundColor: Colors.transparent,
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(20),
        ), // RoundedRectangleBorder
        child: Stack(
          children: [
            Positioned.fill(
              child: Image.asset(
                'assets/icons/light_effect.png',
                opacity: const AlwaysStoppedAnimation(.5),
              ), // Image.asset
            ), // Positioned.fill
            Column(
              mainAxisAlignment: MainAxisAlignment.min,
              children: [
                const Center(
                  child: Text(
                    'Yeey! Dapat Poin...',
                    textAlign: TextAlign.center,
                    style: TextStyle(
                      color: Color(0xFFAAD9BB),
                      fontWeight: FontWeight.bold,
                      fontSize: 30), // TextStyle
                  ), // Text
                ), // Center
                const SizedBox(height: 10),
                Image.asset(
                  'assets/icons/achievement.png',
                  scale: 10,
                ), // Image.asset
                const SizedBox(height: 10),
                Text(
                  "Poin anda +$Poin",
                  style: const TextStyle(
                    color: Color(0xFFAAD9BB),
                    fontWeight: FontWeight.bold,
                    fontSize: 25), // TextStyle
                ), // Text
              ],
            ), // Column
          ],
        ), // Stack
      ); // Dialog
    },
  );
}

```

```

class Leaderboard extends StatefulWidget {
  const Leaderboard({super.key});

  @override
  State<Leaderboard> createState() => _LeaderboardState();
}

class _LeaderboardState extends State<Leaderboard> {
  final ApiConfig _apiClient = ApiConfig();
  late Future<awards.GetAwards> dataAwards;
  late Future<profile.ProfileResponse> dataProfile;

  @override
  void initState() {
    // TODO: implement initState
    dataAwards = _apiClient.getAwards();
    dataProfile = _apiClient.getProfile();
    super.initState();
  }

  int getRankMe(List<awards.Data> listAwards, String userId) {
    int rank = 1;
    for (var data in listAwards) {
      if (data.id == userId) {
        return rank;
      }
      rank++;
    }
    return rank;
  }

  @override
  Widget build(BuildContext context) {
    final double deviceWidth = MediaQuery.of(context).size.width;

    return Scaffold(
      appBar: appBar(context, "Leaderboard", "/profile"),
      body: FutureBuilder(
        // Future Get Profile
        future: dataProfile,
        builder: (context, snapshot) {
          if (snapshot.connectionState == ConnectionState.waiting) {
            return const Center(child: CircularProgressIndicator());
          } else if (snapshot.hasError || !snapshot.hasData) {
            return const Center(
              child: Text(
                'Error: Something went wrong',
                style: TextStyle(
                  color: Colors.red,
                  fontSize: 12.0,
                  fontWeight: FontWeight.w500), // TextStyle
            )); // Text // Center
          } else {
            String userId = snapshot.data!.data!.id!;
            String userName = snapshot.data!.data!.name!;

```



```

int userPoint = snapshot.data!.data!.point!;
return FutureBuilder<awards.GetAwards>(
  // Future Get list Awards
  future: dataAwards,
  builder: (context, snapshot) {
    if (snapshot.connectionState == ConnectionState.waiting) {
      return const Text(
        "Loading ...",
        style: TextStyle(
          fontSize: 15,
          fontWeight: FontWeight.w500,
          color: Colors.white), // TextStyle
      ); // Text
      // return const Center(child: CircularProgressIndicator());
    } else {
      List<awards.Data> listAwards = snapshot.data!.data!;
      int rankMe = getRankMe(listAwards, userId);
      return Column(
        children: [
          // Informasi profile
          Container(
            padding: const EdgeInsets.fromLTRB(20, 10, 0, 10),
            margin: const EdgeInsets.fromLTRB(25, 25, 25, 30),
            width: double.infinity,
            height: 80,
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(15),
              boxShadow: const [
                BoxShadow(
                  color: Colors.black26,
                  blurRadius: 5.0,
                  offset: Offset(2, 4),
                ) // BoxShadow
              ],
              gradient: const LinearGradient(
                colors: [
                  Color(0xFF66ABAC),
                  Color(0xFF8ADBDD)
                ],
                begin: Alignment.bottomLeft,
                end: Alignment.topRight,
              ), // LinearGradient
            ), // BoxDecoration
          child: Column(
            children: [
              Container(
                margin: const EdgeInsets.only(
                  top: 5, bottom: 5), // EdgeInsets.only
                child: const Align(
                  alignment: Alignment.centerLeft,
                  child: Text(
                    "Peringkat-mu",
                    style: TextStyle(

```

```

        style: TextStyle(
          fontSize: 12,
          color: Colors.white,
          fontWeight: FontWeight.bold), // TextStyle
      ), // Text
    ), // Align
  ), // Container
Row(
  children: [
    Container(
      margin:
        const EdgeInsets.only(right: 20),
      width: deviceWidth * 0.45,
      child: Row(
        children: [
          Container(
            width: 20,
            child: Center(
              child: Text(
                rankMe.toString(),
                style: const TextStyle(
                  fontSize: 15,
                  color: Colors.white,
                  fontWeight: FontWeight.w600,
                ), // TextStyle
              ), // Text
            ), // Center
          ), // Container
          Container(
            margin: const EdgeInsets.only(
              left: 20), // EdgeInsets.only
            child: Text(
              userName.length > 12
                ? "${userName.substring(0, 12)}..."
                : userName,
              style: const TextStyle(
                fontSize: 15,
                color: Colors.white,
                fontWeight: FontWeight.w600,
              ), // TextStyle
              overflow: TextOverflow.ellipsis,
              maxLines: 1,
            ), // Text
          ), // Container
        ],
      ), // Row
    ), // Container
    Container(
      width: deviceWidth * 0.3,
      child: Row(
        children: [
          Container(

```

```

        child: Center(
          child: Text(
            userPoint.toString(),
            style: const TextStyle(
              fontSize: 15,
              color: Colors.white,
              fontWeight: FontWeight.w600,
            ), // TextStyle
          ), // Text
        ), // Center
      ), // Container
    Image.asset(
      "assets/icons/coin.png",
      scale: 15,
    ), // Image.asset
  ],
), // Row
), // Container
],
), // Column
), // Container
// List Awards
Expanded(
  child: ListView.builder(
    itemCount: min(listAwards.length, 10),
    itemBuilder: (context, index) {
      String nama = listAwards[index].nama!;
      int poin = listAwards[index].poin!;
      return listRank(
        deviceWidth, index + 1, nama, poin);
    },
  )) // ListView.builder // Expanded
],
); // Column
}
}); // FutureBuilder
}
}); // FutureBuilder // Scaffold
}

// ranked
Container listRank(double deviceWidth, int index, String nama, int poin) {...
```

DAFTAR PUSTAKA

- Dart. (2024, November 11). Retrieved from Dart programming language: <https://dart.dev/>
- Figma. (2024, November 7). *The Collaborative Interface Design Tool*. Retrieved from <https://www.figma.com/>
- Flutter. (2024, November 10). Retrieved from Build apps for any screen: <https://flutter.dev/>
- MDN. (2024, November 11). *Mobile accessibility*. Retrieved from mdn web docs: <https://developer.mozilla.org/en-US/docs/Learn/Accessibility/Mobile>

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS

Nama : Mochammad Taslam Gustino Prianto
Tempat, Tanggal Lahir : Jember, 7 Agustus 2002
Jenis Kelamin : Laki-laki
Telepon : +6289676414445
Email : mtgpmm2013@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2021
Semester : 7 (Tujuh)