

LAPORAN UTS MOBILE PROGRAMMING
"GET FIT GYM" - APLIKASI PANDUAN LATIHAN GYM



Nama : Faril Isra Albiansyah
NIM : 240605110087
Kelas : Mobile Programming B
Tanggal : 20 – 10 – 2025

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
GANJIL 2025/2026

I. Pendahuluan

Seiring dengan meningkatnya kesadaran masyarakat akan pentingnya kesehatan dan kebugaran jasmani, popularitas pusat kebugaran Gym terus meningkat. Namun, banyak pemula yang merasa bingung atau terintimidasi saat pertama kali memulai. Mereka seringkali tidak tahu harus memulai dari mana, latihan apa yang harus dilakukan, atau bagaimana cara melakukan gerakan dengan benar.

Aplikasi mobile "GET FIT GYM" dikembangkan untuk menjawab permasalahan tersebut. Aplikasi ini dirancang sebagai panduan latihan gym yang personal dan mudah diakses, khususnya bagi pengguna pemula hingga menengah.

Pemilihan tema ini didasarkan pada relevansinya dengan kebutuhan pasar saat ini dan potensinya sebagai platform untuk mendemonstrasikan penguasaan konsep-konsep inti pemrograman mobile. Pengembangan aplikasi ini menjadi studi kasus yang ideal untuk menerapkan materi perkuliahan, mulai dari perancangan layout kompleks, navigasi multi-halaman, hingga manajemen data (data dummy) dan state management (GetX), yang semuanya merupakan komponen krusial dalam pengembangan aplikasi mobile modern.

II. Perancangan Antarmuka (UI Design)

Perancangan antarmuka "GET FIT GYM" bertujuan untuk menciptakan pengalaman pengguna yang imersif, modern, dan fungsional.

❖ Konsep Desain

Konsep utama desain aplikasi ini adalah "Dark Mode Fitness". Penggunaan tema gelap (dark mode) dipilih untuk memberikan kesan premium, profesional, dan fokus, serta mengurangi kelelahan mata pengguna saat berlatih di dalam ruangan (gym) yang mungkin memiliki pencahayaan redup.

❖ Palet Warna

Palet warna yang digunakan dirancang agar konsisten dan fungsional:

- Warna Primer (Background): Color(0xFF1C1C1E) (Hitam Pekat) - Memberikan kesan premium dan fokus.
- Warna Sekunder (Kontainer/Kartu): Color(0xFF2C2C2E) (Abu-abu Gelap) - Digunakan untuk semua elemen interaktif seperti kartu dan bottom navigation bar agar terlihat menonjol dari background.
- Warna Aksen: Colors.amber - Digunakan untuk tombol atau tab yang sedang aktif, memberikan kontras yang kuat dan visual yang energik.
- Warna Teks: Colors.white (Teks Utama) dan Colors.grey (Sub-teks/Info Tambahan).

❖ Tata Letak (Layout)

Tata letak aplikasi dirancang untuk menjadi intuitif dan fokus pada konten, mengikuti alur pengguna yang logis:

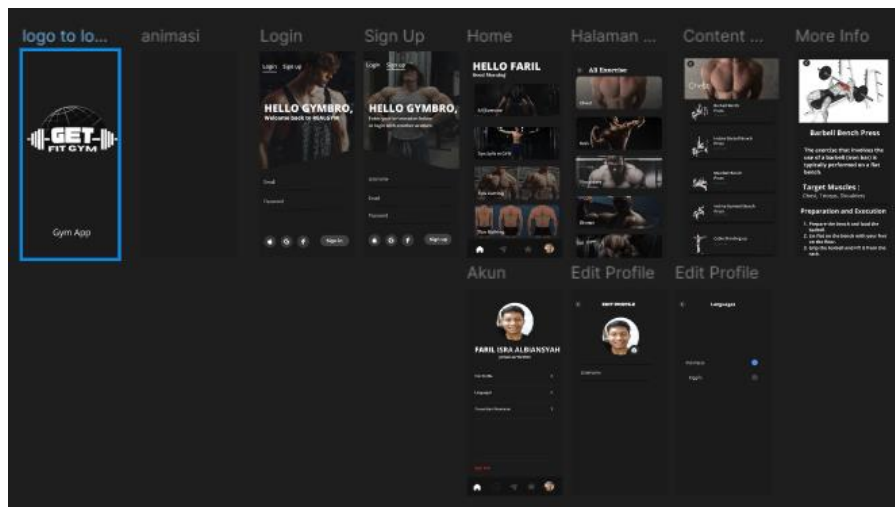
- **Susunan Hierarkis:** Informasi disajikan secara hierarkis. Halaman utama (HomePage) menggunakan daftar vertikal yang dapat digulir untuk menampilkan kartu-kartu kategori utama (All Exercise, Tips, dll). Ini memungkinkan penambahan kategori di masa depan tanpa merusak tata letak.
- **Struktur Halaman (Screen Structure):** Setiap halaman memiliki struktur yang konsisten, terdiri dari App Bar (bilah judul) kustom di bagian atas dan area konten utama di bawahnya. Halaman utama juga dilengkapi dengan bilah navigasi utama di bagian bawah layar (BottomNavigationBar) untuk akses cepat ke Home, Profil, dan fitur lainnya.
- **Desain Kartu (Card Design):** Sebagian besar konten interaktif, seperti kategori latihan dan daftar latihan, disajikan dalam bentuk kartu (cards). Kartu-kartu ini menggunakan gambar latar belakang penuh dengan efek gradient gelap di bagian bawah untuk memastikan teks judul tetap terbaca jelas di atas gambar.
- **Tata Letak Detail:** Halaman detail latihan menggunakan susunan multi-kolom (gambar di kiri, teks judul di kanan) yang diikuti oleh daftar vertikal untuk menampilkan deskripsi, target otot, dan langkah-langkah persiapan secara rapi dan mudah dibaca.

❖ Tipografi

Menggunakan font sans-serif yang modern dan mudah dibaca (seperti Poppins atau font default Material) untuk memastikan keterbacaan di semua ukuran layar. Ukuran dan ketebalan font (FontWeight) digunakan untuk menciptakan hierarki visual (judul, subjudul, konten).

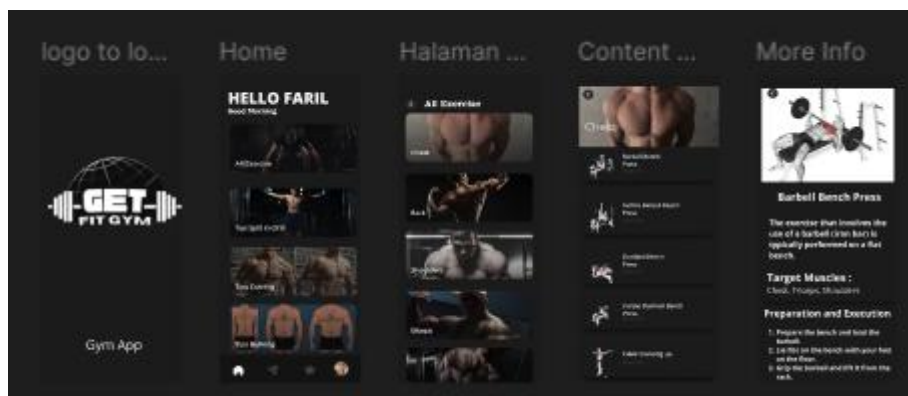
❖ Desain Acuan

Desain antarmuka aplikasi ini diadaptasi dari sebuah prototipe UI/UX yang pernah saya kembangkan semasa SMK. Prototipe tersebut awalnya terinspirasi dari template Gym Fitness App UI Kit yang ada di Komunitas Figma. Untuk proyek saat ini, desain lama tersebut saya kembangkan dan poles kembali agar lebih sesuai dengan alur dan fungsionalitas aplikasi.



Screenshot dari figma yang saya pakai untuk acuan UI proyek saya

Untuk sementara ini dari banyaknya halaman di desain tersebut saya hanya mengambil bagian loading page (logo) , home, halaman all exercise, lalu halaman content all exercise, dan yang terakhir detail contentnya.



Screenshot bagian desain dari figma yang saya ambil.

IV. Implementasi Program

Bagian ini akan memaparkan detail teknis implementasi aplikasi "GET FIT GYM", mulai dari struktur halaman, penggunaan widget, sistem navigasi, hingga cara aplikasi mengolah dan menampilkan data dummy dari file JSON.

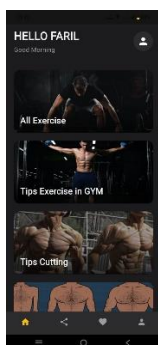
❖ Struktur Halaman (Routes)

Aplikasi ini dikembangkan dengan alur navigasi yang terdiri dari 6 halaman utama, masing-masing memiliki fungsi spesifik:

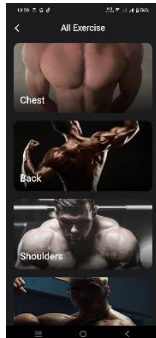
- `loading_page.dart`: Halaman splash screen yang berfungsi sebagai layar pembuka aplikasi. Halaman ini akan tampil selama beberapa detik sebelum secara otomatis mengarahkan pengguna ke halaman utama.



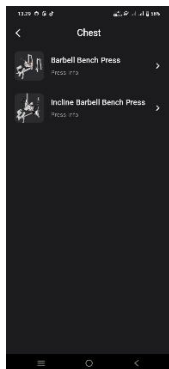
- `home_page.dart`: Halaman utama (dashboard) yang menyambut pengguna dan menjadi pusat navigasi. Halaman ini berisi daftar kategori latihan dan BottomNavigationBar untuk berpindah ke halaman utama lainnya seperti Profil.



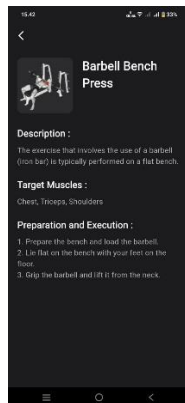
- `focus_exercise_page.dart`: Halaman untuk memilih fokus latihan berdasarkan bagian tubuh. Halaman ini secara dinamis menampilkan daftar bagian tubuh (misal: "Chest", "Back") yang datanya diambil dari controller.



- `list_all_exercise_page.dart`: Halaman yang menampilkan daftar latihan spesifik berdasarkan bagian tubuh yang dipilih dari halaman sebelumnya. Data yang ditampilkan di sini disaring dan dikirim melalui navigasi.



- detail_exercise_page.dart: Halaman akhir yang menampilkan rincian lengkap dari satu gerakan latihan, termasuk deskripsi, target otot, dan cara melakukannya.



- profile_page.dart: Halaman yang berisi informasi profil pengguna dan menu pengaturan. Halaman ini dapat diakses melalui BottomNavigationBar di HomePage.



Cuplikan Kode Penting

Berikut adalah cuplikan kode yang merepresentasikan komponen-komponen inti dari implementasi aplikasi.

1. Pengolahan Data Dummy (Model)

Untuk mengubah data mentah dari file JSON menjadi objek yang terstruktur di dalam Dart, dibuat sebuah model class di `lib/model/dummy_model.dart`. Model ini berfungsi sebagai "cetakan" untuk setiap entitas data.

```
Dart
// Lokasi: lib/model/dummy_model.dart
// Cetakan untuk "Body Part" (Chest, Back, Shoulders)
class BodyPart {
  final String name;
  final String imagePath;
  final List<Exercise> exercises;
  // ... (constructor dan factory fromJson)
}

// Cetakan untuk "Exercise" (Barbell Press, Pull Up)
class Exercise {
  final String name;
  final String imagePath;
  final String info;
  final String description;
  final String targetMuscles;
  final String preparation;
  // ... (constructor dan factory fromJson)
}
```

2. Manajemen State dan Logika (Controller)

Seluruh logika untuk memuat dan mengelola data latihan dipusatkan di `lib/controller/exercise_controller.dart`. Controller ini menggunakan GetX untuk menyediakan data secara reaktif ke view.

Dart

```
// Lokasi: lib/controller/exercise_controller.dart
```

```
class ExerciseController extends GetxController {  
  // Variabel reaktif untuk menampung daftar bagian tubuh  
  var bodyPartList = <BodyPart>[].obs;  
  var isLoading = true.obs;  
  
  @override  
  void onInit() {  
    super.onInit();  
    loadExerciseData(); // Langsung muat data saat controller dibuat  
  }  
  
  // Fungsi untuk membaca dan mem-parsing data dari file JSON  
  Future<void> loadExerciseData() async {  
    try {  
      isLoading.value = true;  
  
      // Membaca file JSON dari assets
```

```

    final String jsonString = await
rootBundle.loadString('assets/dummy_gym_app.json');
    final List<dynamic> jsonData = json.decode(jsonString);

    // Mengubah JSON menjadi List<BodyPart> menggunakan model
    bodyPartList.value = jsonData.map((item) =>
BodyPart.fromJson(item)).toList();

    } finally {
        isLoading.value = false;
    }
}
}

```

Penjelasan: Controller di atas bertanggung jawab penuh atas proses loading data dari assets. View hanya perlu "meminta" data yang sudah matang dari bodyPartList.

3. Menampilkan Data Dinamis (View)

Halaman FocusExercisePage adalah contoh implementasi bagaimana view berinteraksi dengan controller untuk menampilkan data. Get.put() digunakan untuk menginisialisasi controller, dan Obx() digunakan untuk membuat UI menjadi reaktif.

Dart

```
// Lokasi: lib/view/focus_exercise_page.dart
```

```

// ... (kode import)

class FocusExercisePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    // 1. Inisialisasi atau temukan instance controller
    final ExerciseController controller = Get.put(ExerciseController());

    return Scaffold(
      // ... (kode AppBar)

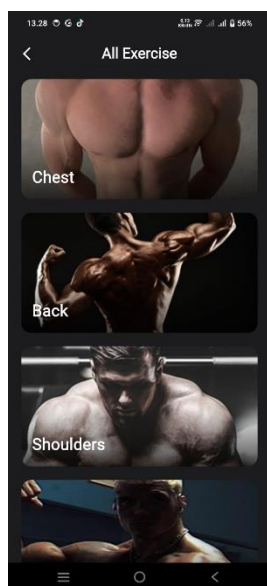
      // 2. Bungkus body dengan Obx agar UI otomatis update
      body: Obx(() {
        // Tampilkan loading spinner jika data belum siap
        if (controller.isLoading.value) {
          return const Center(child: CircularProgressIndicator());
        }

        // 3. Gunakan ListView.builder untuk menampilkan data dinamis
        return ListView.builder(
          itemCount: controller.bodyPartList.length,
          itemBuilder: (context, index) {
            final BodyPart bodyPart = controller.bodyPartList[index];
            return _buildBodyPartCard(
              // ... (data dari 'bodyPart' dimasukkan ke widget kartu)
            );
          },
        );
      }),
    );
  }
}

```

```
);  
}  
// ...  
}
```

Penjelasan: Penggunaan Obx dan ListView.builder memastikan bahwa daftar bagian tubuh ditampilkan secara otomatis setelah controller berhasil memuat data, tanpa perlu setState().



Gambar diatas adalah tampilan halaman FocusExercisePage dengan data dinamis.

4. Navigasi dan Pengiriman Data Antar Halaman

Keterhubungan data antar halaman diimplementasikan dengan mengirim objek atau variabel melalui constructor saat berpindah halaman menggunakan Navigator.push.

Dart

```
// Di dalam `focus_exercise_page.dart`
```

```

// ...
onTap: () {
  // Mengirim data 'bodyPart' ke halaman selanjutnya
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (_) => ListAllExercisePage(
        bodyPartName: bodyPart.name,
        exercises: bodyPart.exercises,
      ),
    ),
  );
},

// Di dalam `list_all_exercise_page.dart`

class ListAllExercisePage extends StatelessWidget {
  // "Pintu" untuk menerima data
  final String bodyPartName;
  final List<Exercise> exercises;

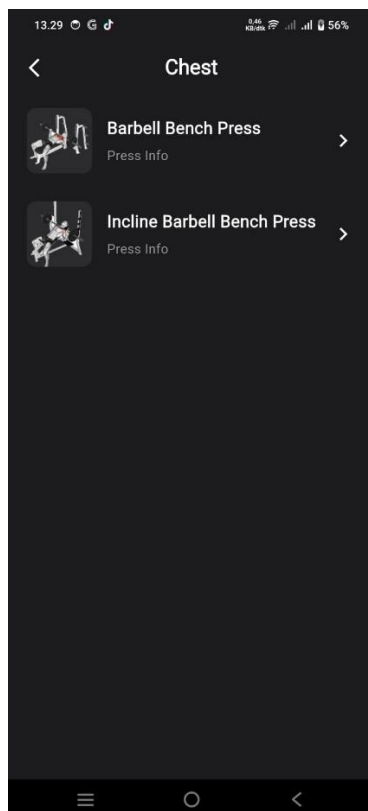
  // Constructor yang wajib menerima data
  const ListAllExercisePage({
    super.key,
    required this.bodyPartName,
    required this.exercises,
  });

  @override
  Widget build(BuildContext context) {

```

```
// ... (widget build menggunakan variabel 'bodyPartName' dan 'exercises')  
}  
}
```

Penjelasan: Cuplikan di atas menunjukkan alur pengiriman data yang jelas. FocusExercisePage mengirimkan data bagian tubuh yang dipilih, dan ListAllExercisePage siap menerimanya untuk kemudian ditampilkan.



Gambar diatas adalah tampilan halaman ListAllExercisePage setelah menerima data "Chest".

IV. Hasil dan Pembahasan

❖ Kesesuaian Desain dengan Implementasi

Hasil implementasi akhir aplikasi "GET FIT GYM" mengambil inspirasi visual utama dari desain acuan di figma dari template Fitness app UI kit, namun tidak mereplikasi keseluruhan desain tersebut secara kaku.

Proyek ini berfokus pada implementasi alur pengguna inti yang paling penting (dari halaman utama, pemilihan bagian tubuh, hingga detail latihan). Keputusan untuk tidak mengimplementasikan semua halaman dari acuan diambil agar pengembangan lebih fokus dan sesuai dengan lingkup pengerjaan Ujian Tengah Semester.

Selain itu, dilakukan beberapa modifikasi desain dari acuan aslinya. Perubahan ini bertujuan untuk menyederhanakan beberapa komponen dan menyesuaikan tata letak agar lebih fungsional dan nyaman dipandang mata (enak dipandang), sekaligus memastikan aplikasi dapat berjalan dengan baik sesuai dengan data dummy yang telah disiapkan.

Meskipun demikian, konsep visual inti dari desain acuan tersebut berhasil dipertahankan dan diadaptasi dengan baik:

- Palet Warna dan Tema: Implementasi tetap mengadopsi tema dark mode yang konsisten dengan warna primer 0xFF1C1C1E (background), sekunder 0xFF2C2C2E (kartu), dan aksen Colors.amber. Pilihan palet warna ini berhasil menciptakan nuansa premium dan fokus seperti yang diharapkan.
- Tata Letak (Layout): Struktur visual telah diinterpretasikan dan disesuaikan dari acuan. Penggunaan ListView pada HomePage dan FocusExercisePage adalah adaptasi fungsional untuk menampilkan daftar kartu vertikal. Demikian pula, halaman detail (DetailExercisePage) dimodifikasi menjadi

tata letak multi-kolom yang lebih ringkas untuk menampilkan gambar, judul, dan teks informatif.

- **Komponen UI:** Komponen kunci seperti BottomNavigationBar dan header "HELLO FARIL" diimplementasikan dengan mempertahankan estetika dari desain acuan, namun dengan penyesuaian minor untuk memastikan konsistensi visual di seluruh aplikasi.

Singkatnya, hasil akhir aplikasi ini adalah sebuah adaptasi fungsional dari desain acuan, di mana prioritas diberikan pada penerapan alur logika dan fungsionalitas inti, dengan tetap mempertahankan identitas visual modern dari desain tersebut.

❖ Fungsi Halaman dan Alur Interaksi Pengguna

Aplikasi ini memiliki alur interaksi pengguna (user flow) yang logis dan terstruktur, yang sepenuhnya didukung oleh data dinamis dari file JSON. Berikut adalah rincian fungsionalitas dan alur kerjanya:

- **Pembukaan Aplikasi:** Pengguna pertama kali disambut oleh LoadingPage yang menampilkan logo dan nama aplikasi. Setelah beberapa detik, aplikasi secara otomatis bernavigasi ke HomePage menggunakan Navigator.pushReplacement untuk memastikan pengguna tidak dapat kembali ke splash screen.
- **Navigasi Utama (HomePage):** Halaman ini berfungsi sebagai dashboard utama. Pengguna dapat melihat daftar kategori latihan utama. Di bagian bawah, BottomNavigationBar menyediakan akses cepat ke halaman utama lainnya, seperti ProfilePage.

- Memulai Alur Latihan: Saat pengguna menekan kartu "All Exercise" di HomePage, aplikasi melakukan navigasi ke FocusExercisePage.
- Pemilihan Fokus Latihan (FocusExercisePage): Di halaman ini, aplikasi secara otomatis memanggil ExerciseController. Controller kemudian memuat data dari file dummy_gym_app.json. Berkat Obx() dari GetX, halaman ini secara reaktif menampilkan daftar bagian tubuh ("Chest", "Back", dll.) yang berhasil dimuat oleh controller.
- Melihat Daftar Latihan (ListAllExercisePage): Ketika pengguna memilih satu bagian tubuh (misalnya, "Chest"), aplikasi melakukan navigasi ke ListAllExercisePage. Pada proses ini, data spesifik (yaitu bodyPart.name dan bodyPart.exercises) dikirimkan melalui constructor halaman. Halaman ini kemudian menampilkan daftar latihan yang relevan dengan bagian tubuh tersebut.
- Mempelajari Detail Latihan (DetailExercisePage): Saat pengguna memilih satu latihan (misalnya, "Barbell Bench Press"), aplikasi kembali melakukan navigasi ke DetailExercisePage, kali ini mengirimkan satu objek Exercise yang dipilih. Halaman ini kemudian menampilkan semua rincian dari objek tersebut, seperti deskripsi, target otot, dan persiapan.

Alur ini menunjukkan bahwa sistem navigasi dan keterhubungan data antar halaman telah berhasil diimplementasikan sesuai rancangan. Setiap halaman memiliki fungsi yang jelas dan bekerja secara sinergis untuk memandu pengguna.

❖ Pencapaian Tujuan Aplikasi

Tujuan utama pengembangan aplikasi "GET FIT GYM" adalah untuk menciptakan sebuah panduan latihan gym yang fungsional, intuitif, dan menarik secara visual, sekaligus menerapkan konsep-konsep inti dari mata kuliah Pemrograman Mobile. Berdasarkan hasil akhir, dapat disimpulkan bahwa tujuan-tujuan tersebut telah berhasil tercapai.

- Aplikasi sebagai Panduan: Alur yang terstruktur dari pemilihan bagian tubuh hingga detail gerakan berhasil menyediakan panduan yang jelas bagi pengguna.
- Implementasi Konsep Teknis: Proyek ini berhasil mendemonstrasikan penguasaan konsep-konsep kunci, termasuk layout kompleks, navigasi dengan pengiriman data, pengolahan data JSON melalui model, dan pemisahan logika menggunakan controller (GetX).
- Pengalaman Pengguna: Antarmuka yang sesuai dengan desain modern dan alur yang logis berhasil menciptakan pengalaman pengguna yang baik dan tidak membingungkan.

Secara keseluruhan, aplikasi ini tidak hanya berfungsi sesuai rancangan, tetapi juga menjadi bukti implementasi yang solid dari materi-materi yang telah dipelajari.

V. Kesimpulan

Proyek Ujian Tengah Semester ini telah berhasil diselesaikan dengan pengembangan aplikasi "GET FIT GYM", sebuah panduan latihan gym berbasis Flutter.

Pencapaian utama dari proyek ini adalah keberhasilan dalam mengimplementasikan seluruh konsep inti dari mata kuliah Pemrograman Mobile, meliputi:

Perancangan dan implementasi layout UI yang kompleks dan responsif menggunakan berbagai widget Flutter.

Penerapan navigasi multi-halaman yang logis (5 halaman) dengan mekanisme pengiriman data antar halaman melalui constructor.

Pengolahan data dummy dari file JSON eksternal, yang di-parsing menggunakan model class (arsitektur Model).

Implementasi arsitektur yang bersih (MVC-like) dengan memisahkan logika bisnis ke dalam Controller (GetX).

Potensi pengembangan lanjutan untuk aplikasi ini sangat terbuka, seperti mengintegrasikan aplikasi dengan database dan API server sungguhan (misalnya Firebase/REST API) untuk menggantikan data dummy, menambahkan fitur autentikasi pengguna, dan mengembangkan fitur tracker progres latihan.