

Studi Kasus

Judul

e-Penilaian Santri Pesantren (Flutter)

Latar Belakang

Pesantren X ingin mendigitalisasi penilaian santri (akademik & non-akademik) agar rapor bisa dilihat wali santri dan pimpinan. Penilaian meliputi Tahfidz, Fiqh, Bahasa Arab, Akhlak/Adab, dan Kehadiran. Sistem harus bekerja **offline-first** dan **sinkron** saat online.

Tugas Utama

Bangun aplikasi Flutter (Android/iOS; opsional Web) dengan fitur inti berikut:

1. Autentikasi & Peran

- Peran: **Admin, Ustadz/Wali Kelas, Wali Santri (read-only)**.
- Minimal: login dengan email+password (mock), simpan token lokal.

2. Data Master

- CRUD **Santri** (NIS, nama, kamar/asrama, angkatan).
- CRUD **Mata Pelajaran/Aspek**: Tahfidz, Fiqh, Bahasa Arab, Akhlak, Kehadiran.
- **Rubrik Akhlak** (skala 1–4: Kurang–Sangat Baik) dengan indikator: Disiplin, Adab pada guru, Kebersihan, Kerja sama.

3. Input Penilaian

- **Tahfidz**: target hafalan (surah/ayat), setoran mingguan (ayat), nilai ketepatan tajwid (0–100).
- **Fiqh & Bahasa Arab**: nilai formatif (0–100) dan sumatif (0–100).
- **Akhlik**: skor per indikator (1–4), rata-rata → 0–100.
- **Kehadiran**: hadir/sakit/izin/alpa per pertemuan → rekap persentase.

4. Perhitungan Nilai Akhir (per santri)

- Bobot default (dapat diubah Admin):
 - Tahfidz 30%
 - Fiqh 20%

- Bahasa Arab 20%
 - Akhlak 20%
 - Kehadiran 10%
- **Konversi Akhlak:** rata-rata indikator (1–4) → skala 100: nilai = $(\text{avg}/4)*100$.
 - **Kehadiran:** hadir% langsung jadi nilai 0–100.
 - **Predikat:**
 - A: 85–100
 - B: 75–84
 - C: 65–74
 - D: <65

5. Rapor

- Tampilkan ringkas: nilai per aspek, total, predikat, catatan ustadz.
- Ekspor **PDF** (opsional HTML) & bagikan.
- Grafik sederhana (bar/line) perkembangan Tahfidz mingguan.

6. Offline-First & Sinkronisasi

- Simpan data lokal (**sqflite** / **drift**) + antrian perubahan.
- Saat online, **sync** ke server mock (lihat API palsu di bawah).
- Tampilkan status sinkron (ikon/label per item).

7. Keamanan Data

- Simpan token/credential dengan **flutter_secure_storage**.
- Batasi akses berdasarkan peran.

Batasan Teknis (Wajib)

- Flutter stable (3.x+), null-safety.
- **State management:** pilih satu (Riverpod/BLoC/Cubit/Provider) dan konsisten.

- **Arsitektur:** **Clean Architecture** (data/domain/presentation) atau MVVM.
- **Lint aktif, null safety, unit test minimal untuk:**
 - Mapper/konversi Akhlak → 0–100
 - Perhitungan nilai akhir & predikat
 - Rekap kehadiran
- **Responsif** (mobile potret), aksesibilitas dasar (font scalable).

Skema Data (contoh)

Entity (simplifikasi)

erDiagram

```

SANTRI ||--o{ PENILAIAN_TAHFIDZ : has
SANTRI ||--o{ PENILAIAN_MAPEL : has
SANTRI ||--o{ PENILAIAN_AKHLAK : has
SANTRI ||--o{ KEHADIRAN : has

```

```

SANTRI {
    String id
    String nis
    String nama
    String kamar
    int angkatan
}

```

```

PENILAIAN_TAHFIDZ {
    String id
    String santrild
    Date minggu
    String surah
}

```

```
int ayat_setor  
int tajwid_0_100  
}
```

```
PENILAIAN_MAPEL {  
    String id  
    String santrild  
    String mapel // "Fiqh" | "Bahasa Arab"  
    int formatif_0_100  
    int sumatif_0_100  
}
```

```
PENILAIAN_AKHLAK {  
    String id  
    String santrild  
    int disiplin_1_4  
    int adab_1_4  
    int kebersihan_1_4  
    int kerjasama_1_4  
    String catatan  
}
```

```
KEHADIRAN {  
    String id  
    String santrild  
    Date tanggal  
    String status // "H","S","I","A"  
}
```

Rumus Perhitungan (detil)

- **Nilai Tahfidz** (0–100):
 - Komponen:
 - Capaian setoran (dibanding target mingguan): $\min(100, (\text{setor}/\text{target}) * 100)$
 - Tajwid (0–100)
 - Gabungan: $\text{tahfidz} = \text{round}(0.5 * \text{capaian} + 0.5 * \text{tajwid})$
- **Fiqh/Bahasa Arab**:
 - $\text{mapel} = \text{round}(0.4 * \text{formatif} + 0.6 * \text{sumatif})$
- **Akhlik (1–4)**:
 - $\text{avg} = \text{mean}([\text{disiplin}, \text{adab}, \text{kebersihan}, \text{kerjasama}])$
 - $\text{akhlak} = \text{round}((\text{avg}/4) * 100)$
- **Kehadiran**:
 - $\text{hadir\%} = (\text{H} / (\text{H+S+I+A})) * 100$
 - $\text{kehadiran} = \text{round}(\text{hadir\%})$
- **Nilai Akhir**:
- $\text{final} = \text{round}($
- $0.30 * \text{tahfidz} +$
- $0.20 * \text{fiqh} +$
- $0.20 * \text{bahasaArab} +$
- $0.20 * \text{akhlak} +$
- $0.10 * \text{kehadiran}$
- $)$
- **Predikat**: sesuai tabel pada fitur (A/B/C/D).

Mock API (untuk sinkron)

Gunakan **REST mock** lokal (contoh dengan json_server/Express) atau simulasikan service:

GET /api/santri

POST /api/santri

GET /api/santri/{id}

GET /api/penilaian/{santrilId}

POST /api/penilaian/sync // kirim batch perubahan offline

GET /api/config/bobot // kembalikan bobot

Contoh respons GET /api/config/bobot:

```
{ "tahfidz": 0.30, "fiqh": 0.20, "bahasaArab": 0.20, "akhlak": 0.20, "kehadiran": 0.10 }
```

Contoh Data Uji (minimal)

Santri

[

```
{"id":"s1","nis":"2025-001","nama":"Ahmad F","kamar":"A3","angkatan":2023},
```

```
{"id":"s2","nis":"2025-002","nama":"Bilal S","kamar":"B1","angkatan":2022}
```

]

Nilai (ringkas)

- Tahfidz (minggu ini; target 50 ayat):
 - s1: ayat_setor=60, tajwid=85 → capaian=100 → tahfidz=93
 - s2: ayat_setor=30, tajwid=78 → capaian=60 → tahfidz=69
- Fiqh:
 - s1: formatif 80, sumatif 90 → 86
 - s2: formatif 70, sumatif 75 → 73
- Bahasa Arab:
 - s1: 75 & 80 → 78
 - s2: 60 & 65 → 63

- Akhlak (1–4):
 - s1: [4,4,3,4] avg=3.75 → 94
 - s2: [3,3,3,2] avg=2.75 → 69
- Kehadiran (20 pertemuan):
 - s1: H=18,S=1,I=1,A=0 → 90
 - s2: H=14,S=2,I=1,A=3 → 70

Nilai Akhir (s1)

$$\begin{aligned}
 &= 0.30*93 + 0.20*86 + 0.20*78 + 0.20*94 + 0.10*90 \\
 &= 27.9 + 17.2 + 15.6 + 18.8 + 9 = 88.5 \rightarrow 89 \text{ (Predikat A)}
 \end{aligned}$$

Nilai Akhir (s2)

$$\begin{aligned}
 &= 0.30*69 + 0.20*73 + 0.20*63 + 0.20*69 + 0.10*70 \\
 &= 20.7 + 14.6 + 12.6 + 13.8 + 7 = 68.7 \rightarrow 69 \text{ (Predikat C)}
 \end{aligned}$$

Tampilan Minimum (wireframe singkat)

1. **Login** (role-based).
2. **Daftar Santri** (search, filter kamar/angkatan, status sinkron).
3. **Detail Santri**: tab *Penilaian, Kehadiran, Grafik Tahfidz, Rapor.*
4. **Form Input**:
 - Tahfidz (surah, ayat, tajwid).
 - Mapel (formatif, sumatif).
 - Akhlak (empat indikator).
 - Kehadiran (H/S/I/A).
5. **Rapor**: kartu nilai + tombol **Unduh PDF**.

Kriteria Penerimaan (Acceptance Criteria)

- Bisa login & membedakan peran (UI/fitur yang terlihat).
- CRUD Santri bekerja offline; data persisten setelah aplikasi ditutup.
- Input penilaian lengkap & tervalidasi (angka, rentang, wajib isi).
- Perhitungan akhir & predikat tepat sesuai rumus (uji otomatis lulus).

- Rapor muncul dan dapat diekspor PDF.
 - Status sinkron terlihat; saat internet kembali, data terkirim (simulasi).
 - Minimal 5 widget test / unit test lulus (service/perhitungan/formatter).
 - Kode rapi, dipisah layer, ada README cara run & test.
-

Penilaian (Rubrik)

Aspek	Bobot	Indikator
Fungsionalitas	35%	Semua fitur inti terpenuhi & bebas bug mayor
Arsitektur & Kualitas Kode	20%	Clean/MVVM, modular, DI, lint
UX/UI	10%	Konsisten, mudah dipakai, responsif
Offline & Sync	15%	Data lokal & mekanisme retry/merge
Keamanan Dasar	5%	Secure storage token, guard peran
Testing	10%	Unit/widget test untuk rumus & UI penting
Dokumentasi	5%	README, diagram singkat, instruksi

Ekstra Poin (Opsional)

- Multi-bahasa (ID/EN/AR).
- Grafik progres Tahfidz multi-minggu.
- Import/Export CSV.
- Dark mode.
- Notifikasi lokal untuk setoran Tahfidz.

Petunjuk Singkat Implementasi

- Paket yang disarankan: flutter_riverpod/bloc, dio, freezed+json_serializable, drift/sqlite, go_router, flutter_secure_storage, printing/pdf, intl, connectivity_plus.

- Struktur folder (contoh Clean):

```
lib/  
core/  
features/  
auth/...  
santri/...  
penilaian/...  
rapor/...  
shared/...
```

Yang Dikumpulkan

1. Link repo (public) + commit history wajar.
2. APK debug/release.
3. README: arsitektur, paket, cara run, cara uji, asumsi.
4. Cuplikan layar (3–5) & contoh PDF rapor.

Kalau mau, aku bisa bantu bikin **template proyek Flutter** awal (folder, model Freezed, dan contoh unit test perhitungan) — tinggal bilang “buatkan templatenya.”