

Nama : Muhammad Alfin Khaerudin

NIM : 1313621003

Prodi : Ilmu Komputer 2021

Mata Kuliah : Mobile Computing

Dokumentasi Tertulis

A. Latar Belakang Pembuatan Aplikasi

Sebagai seorang pelajar yang memiliki pengalaman langsung dalam menghadapi kompleksitas jadwal pelajaran dan tuntutan tugas, saya merasakan dorongan pribadi untuk menciptakan solusi yang dapat mengatasi permasalahan yang saya hadapi sehari-hari. Melalui pengalaman pribadi ini, saya ingin merasakan bagaimana memiliki aplikasi yang dikembangkan secara mandiri untuk mengatasi permasalahan yang konteksnya merupakan masalah diri saya sendiri.

Dalam perjalanan pendidikan saya, saya sering merasa kesulitan mengelola waktu dengan efektif dan terkadang menghadapi keterlambatan karena kesulitan mengingat jadwal pelajaran yang padat. Apalagi di era modern pendidikan saat ini, kebutuhan akan alat bantu pengelolaan waktu dan informasi menjadi semakin penting, terutama bagi para pelajar dan mahasiswa yang memiliki jadwal yang beragam. Kesadaran akan tantangan ini mendorong saya untuk mengeksplorasi potensi pengembangan aplikasi yang dapat memberikan solusi konkret.

Japel.edu bukan hanya sebuah proyek pengembangan aplikasi mobile, tetapi juga sebuah perjalanan pribadi yang memungkinkan saya merasakan bagaimana mengatasi tantangan yang saya hadapi. Melalui pengembangan aplikasi ini, saya belajar mengidentifikasi kebutuhan pribadi dan mengubahnya menjadi sebuah solusi teknologi yang dapat bermanfaat bagi diri saya sendiri dan mungkin juga bagi teman-teman sesama pelajar.

Saya berharap, dengan hadirnya aplikasi japel.edu ini, dapat memberikan dampak positif dan memberdayakan pelajar lainnya dalam mengelola waktu mereka dengan lebih baik. Japel.edu menjadi bukti bahwa solusi untuk permasalahan pribadi dapat menjadi kontribusi berarti dalam pengembangan teknologi yang bersifat inklusif dan memberikan nilai tambah kepada pelajar.

B. Penjelasan Fitur-Fitur

a. Splash Screen

Kondisi pertama kali saat aplikasi dibuka dan dia akan melakukan seperti namanya yaitu splash.



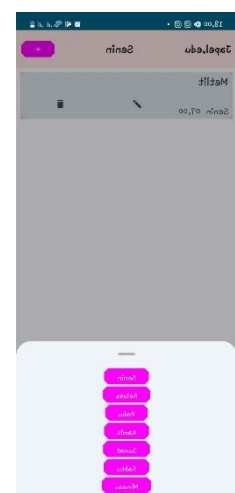
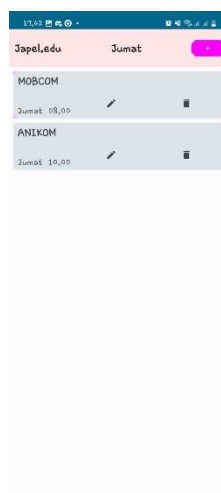
b. Start Screen

Halaman pertama aplikasi saat dijalankan. Halaman ini berisi nama, logo, dan pengembang dari aplikasi japel.edu.



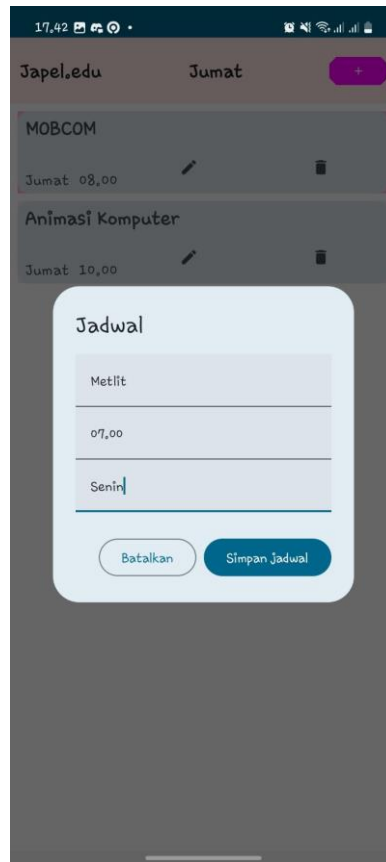
c. Main Screen

Halaman utama dari aplikasi japel.edu. Halaman ini terdiri dari screen yang sudah ada jadwalnya maupun yang belum. Disini juga saya mengadopsi fitur modalbuttonsheet yang dimiliki oleh compose untuk menyimpan semua hari yang ada. Hari yang ada di modal tersebut nantinya akan terfilter secara otomatis saat kita milih hari tertentu.



d. Add Jadwal

Halaman untuk menambahkan jadwal.



17,42

Japel.edu Jumat +

MOBCOM

Jumat 08,00

Animasi Komputer

Jumat 10,00

Jadwal

Metlit

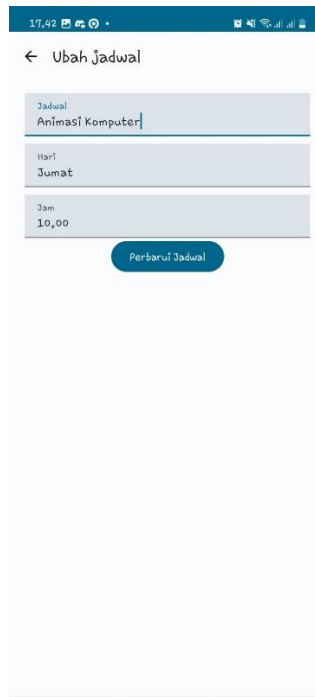
07,00

Senin

Batalan Simpan Jadwal

e. Edit Jadwal

Halaman untuk mengedit jadwal.



17,42

← Ubah Jadwal

Jadwal
Animasi Komputer

Hari
Jumat

Jam
10,00

Perbarui Jadwal

f. Delete Jadwal

Untuk menghapus jadwal saya akan memberikan baris kode dari perlakuan hapus jadwal. Perlakuan delete tersebut ada di file MainViewModel.kt yang nanti diambil dari ViewModel.kt untuk menjalankan modelnya. Lalu memanggil val repository yang isinya merupakan perlakuan dari file NoteRepository.kt

Salah satu fungsi dari file MainViewModel.kt

```
MalfinK
fun deleteNote(note: Note) {
    viewModelScope.launch(Dispatchers.IO) { this: CoroutineScope
        repository.deleteNote(note = note)
    }
}
```

Isi file ViewModel.kt

```
6
7 package androidx.lifecycle
8
9 import ...
5
6 private const val JOB_KEY = "androidx.lifecycle.ViewModelCoroutineScope.JOB_KEY"
7
8 CoroutineScope tied to this ViewModel. This scope will be canceled when ViewModel will be cleared, i.e
9 ViewModel.onCleared is called
10 This scope is bound to Dispatchers.Main.immediate
5 public val ViewModel.viewModelScope: CoroutineScope
6 get() {
7     val scope: CoroutineScope? = this.getTag(JOB_KEY)
8     if (scope != null) {
9         return scope
10    }
11    return setTagIfAbsent(
12        JOB_KEY,
13        CloseableCoroutineScope(context: SupervisorJob() + Dispatchers.Main.immediate)
14    )
15 }
6
7 internal class CloseableCoroutineScope(context: CoroutineContext) : Closeable, CoroutineScope {
8     override val coroutineContext: CoroutineContext = context
9
10    override fun close() {
11        coroutineContext.cancel()
12    }
13 }
14 }
```

Salah satu fungsi dari file NoteRepository.kt

```
MalfinK
suspend fun updateNote(note: Note): Unit = dao.updateNote(note = note)
```