



PENGEMBANGAN APLIKASI COURSETASK UNTUK MENDUKUNG MANAJEMEN TUGAS DAN INFORMASI MATA KULIAH BERBASIS FLUTTER

Vergino F. Maindoka ¹, Meylivia J. M. Liuw ², Andrew Evert Daniel Tondang ³, Ade Yusupa ⁴, Victor Tarigan ⁵

^{1,2,3,4,5}Teknik Informatika, Fakultas Teknik, Universitas Sam Ratulangi

verginomaindoka026@student.unsrat.ac.id ¹, meylivialiuw026@student.unsrat.ac.id ²,
andrewtondang026@student.unsrat.ac.id ³, ade@unsrat.ac.id ⁴, victortarigan@unsrat.ac.id ⁵

ABSTRACT

This study presents the development of CourseTask, a mobile application built with Flutter to help students manage academic tasks and course-related information more effectively. The application provides task management features equipped with readable priority and status labels, and allows tasks to be linked to specific courses or recorded independently as general tasks. Users can also store course information such as instructor names, schedules, and brief descriptions, enabling clearer context for their academic activities. All data is stored locally using SQLite through the Drift library, allowing the application to function without an internet connection.

The development process follows the Waterfall model, which includes requirement analysis, system design, implementation, and functional testing. The results demonstrate that the core features operate reliably and that the clean, simple interface helps users quickly understand important information. CourseTask successfully meets its goal as a practical and easy-to-use academic management tool, and it offers potential for further enhancement through the addition of features such as reminders or cross-device synchronization.

Keywords: *task management, mobile application, Flutter, Riverpod, Drift, SQLite*

ABSTRAK

Penelitian ini membahas pengembangan CourseTask, sebuah aplikasi mobile berbasis Flutter yang dirancang untuk membantu mahasiswa mengelola tugas dan informasi mata kuliah secara lebih teratur. Aplikasi ini menyediakan fitur pencatatan tugas dengan status dan prioritas yang ditampilkan dalam bentuk label yang mudah dipahami, serta memungkinkan pengguna untuk menautkan tugas dengan mata kuliah tertentu maupun mencatat tugas umum yang tidak terkait dengan mata kuliah. Informasi mata kuliah seperti nama pengajar, jadwal, dan deskripsi singkat juga dapat disimpan sehingga pengguna memperoleh konteks yang lebih jelas terkait aktivitas perkuliahan mereka. Seluruh

Article History

Received:

Reviewed:

Published:

Plagiarism Checker No

Prefix DOI :

Copyright : Author

Publish by : Kohesi



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/)



data disimpan secara lokal menggunakan SQLite melalui pustaka Drift, sehingga aplikasi tetap dapat digunakan tanpa koneksi internet.

Pengembangan aplikasi dilakukan dengan model Waterfall yang mencakup tahap analisis kebutuhan, perancangan, implementasi, dan pengujian fungsional. Hasil pengembangan menunjukkan bahwa fitur-fitur utama dapat berjalan dengan baik dan antarmuka yang dirancang sederhana membantu pengguna memahami informasi penting secara cepat. CourseTask berhasil memenuhi tujuan pengembangannya sebagai aplikasi yang praktis dan mudah digunakan, serta memiliki potensi untuk dikembangkan lebih lanjut melalui penambahan fitur seperti pengingat atau sinkronisasi data lintas perangkat.

Kata Kunci: manajemen tugas, aplikasi mobile, Flutter, Riverpod, Drift, SQLite

1. PENDAHULUAN

Perkembangan teknologi mobile membuka kesempatan bagi mahasiswa untuk mengatur aktivitas kuliah dengan cara yang lebih praktis. Namun, aplikasi pencatat tugas yang tersedia saat ini umumnya dirancang untuk penggunaan umum sehingga tidak benar-benar menyesuaikan dengan kebutuhan khas dunia perkuliahan. Pada kenyataannya, mahasiswa tidak hanya berurusan dengan daftar tugas, tetapi juga berbagai informasi seputar mata kuliah yang mereka ikuti—hal-hal yang berkaitan dengan perkuliahan sehari-hari dan sangat mempengaruhi cara mereka menentukan prioritas.

Situasi inilah yang mendorong pengembangan sebuah aplikasi yang bukan hanya mencatat tugas, tetapi juga mengaitkannya dengan mata kuliah yang relevan. Dengan cara ini, mahasiswa bisa mencatat tugas yang berdiri sendiri maupun tugas yang terhubung langsung dengan mata kuliah tertentu. Aplikasi juga menyediakan ruang untuk menyimpan informasi dasar mata kuliah seperti identitas dosen, kelas yang diikuti, dan jadwal perkuliahan. Pendekatan ini membuat manajemen tugas terasa lebih menyatu dengan konteks perkuliahan, sehingga mahasiswa dapat memahami gambaran akademiknya dengan lebih jelas.

Untuk mewujudkan aplikasi ini, Flutter dipilih sebagai kerangka kerja utama karena kematangan ekosistemnya, performa yang baik, dan fleksibilitas dalam membangun antarmuka modern. Data aplikasi disimpan secara lokal menggunakan SQLite melalui pustaka Drift, yang memungkinkan pengelolaan data yang rapi dan tetap



bekerja tanpa koneksi internet. Manajemen status menggunakan Riverpod, yang menawarkan pola kerja yang bersih, terstruktur, dan mudah dikembangkan ketika aplikasi diperluas. Kombinasi teknologi tersebut memberikan fondasi yang kuat untuk menghadirkan aplikasi yang stabil dan nyaman digunakan.

Penelitian ini berfokus pada proses pengembangan aplikasinya, mulai dari tahap analisis kebutuhan hingga pengujian. Pembahasan diarahkan lebih banyak pada bagaimana aplikasi ini dibangun dari sisi teknis, sementara aspek fungsional aplikasi tetap dijelaskan untuk memberikan konteks penggunaannya. Harapannya, hasil penelitian ini tidak hanya menghasilkan aplikasi yang bermanfaat bagi mahasiswa, tetapi juga dapat menjadi referensi pengembangan aplikasi mobile yang memanfaatkan Flutter dan teknologi pendukungnya secara optimal.

2. METODE PENELITIAN

Penelitian ini menggunakan model pengembangan sistem berurutan melalui pendekatan Waterfall. Model ini dipilih karena kebutuhan aplikasi telah ditentukan sejak awal dan perubahan selama proses pengembangan relatif kecil. Setiap tahap dalam model ini memberikan keluaran yang menjadi dasar bagi tahap berikutnya sehingga proses dapat berjalan terarah dan konsisten.

Tahap pertama adalah analisis kebutuhan. Pada tahap ini, permasalahan yang ingin diselesaikan dipetakan, yaitu kebutuhan mahasiswa untuk mencatat tugas serta informasi mata kuliah secara sederhana dan terpadu. Hasil analisis kebutuhan menggambarkan bagaimana aplikasi seharusnya membantu mahasiswa, fitur apa saja yang diperlukan, serta batasan ruang lingkup sistem yang akan dikembangkan.

Tahap berikutnya adalah perancangan, yang berfokus pada desain struktur sistem. Pada tahap ini ditentukan teknologi yang akan digunakan, alur navigasi aplikasi, struktur basis data, serta rancangan antarmuka pengguna. Pemilihan Flutter sebagai kerangka kerja pengembangan mobile dilakukan pada tahap ini karena kemampuannya menghasilkan antarmuka yang konsisten dan cepat dikembangkan. Penggunaan Drift untuk basis data lokal dipilih karena mendukung SQLite secara terstruktur, sedangkan Riverpod dipilih untuk manajemen status karena kesederhanaan, fleksibilitas, dan kemudahan pemeliharaan. Tahap perancangan juga mencakup pembuatan diagram sederhana untuk menggambarkan hubungan antar fitur dan bagaimana data akan dikelola di dalam aplikasi.



Tahap implementasi merupakan proses menerjemahkan rancangan ke dalam bentuk aplikasi yang berfungsi. Struktur kode mulai dibangun berdasarkan arsitektur yang telah ditentukan pada tahap sebelumnya. Setiap fitur—seperti pencatatan tugas, pengelolaan mata kuliah, serta hubungan antara keduanya—direalisasikan secara bertahap dan diuji secara lokal untuk memastikan berjalan sesuai rancangan. Pada tahap ini, fokus utamanya adalah konsistensi struktur kode, kejelasan modul, dan penerapan desain antarmuka yang telah dibuat.

Setelah implementasi selesai, dilakukan pengujian. Pengujian dilakukan secara fungsional untuk memastikan bahwa semua fitur dapat digunakan dengan benar: mulai dari menambah, mengubah, dan menghapus tugas hingga menyimpan informasi mata kuliah dan menampilkan keterkaitan antara keduanya. Pengujian juga memastikan bahwa alur navigasi berjalan mulus dan data tersimpan dengan benar di basis data lokal. Tahap ini penting untuk menemukan kesalahan logika maupun masalah teknis sebelum aplikasi dinyatakan siap.

Tahap terakhir adalah pemeliharaan, yang mencakup evaluasi dan penyempurnaan aplikasi berdasarkan hasil pengujian. Meskipun penelitian ini tidak fokus pada pengembangan jangka panjang, tahap pemeliharaan tetap dilakukan untuk memastikan aplikasi berada dalam kondisi yang stabil dan penggunaannya tidak mengalami kendala. Perbaikan kecil, penyesuaian antarmuka, maupun optimalisasi struktur kode termasuk dalam bagian ini.

Dengan mengikuti alur Waterfall dari awal hingga akhir, pengembangan aplikasi dapat berlangsung secara terstruktur, mulai dari identifikasi kebutuhan hingga penyempurnaan hasil akhir. Pendekatan ini memberikan kerangka kerja yang jelas sehingga seluruh proses dapat dijelaskan dan dipertanggungjawabkan secara sistematis.

3. HASIL DAN PEMBAHASAN

Bagian ini menjelaskan hasil pengembangan aplikasi CourseTask beserta pemanfaatan fitur-fiturnya. Pembahasan meliputi gambaran umum aplikasi, analisis kebutuhan melalui use case, perancangan basis data, implementasi fitur utama, serta pengujian fungsional untuk memastikan aplikasi berjalan sebagaimana mestinya.



3.1 Gambaran Umum Aplikasi

CourseTask dikembangkan sebagai aplikasi mobile yang membantu mahasiswa mengelola tugas dan informasi mata kuliah secara lebih terorganisir. Aplikasi ini menyediakan tiga halaman utama: Dashboard, Assignments, dan Lectures. Dashboard menampilkan ringkasan tugas seperti jumlah total, tugas yang masih terbuka, tugas yang sudah selesai, serta daftar deadline terdekat. Halaman Lectures berfungsi untuk menyimpan informasi mata kuliah yang diikuti pengguna, termasuk nama pengajar, jadwal perkuliahan, dan deskripsi singkat jika diperlukan. Sementara itu, halaman Assignments menjadi pusat pengelolaan seluruh tugas yang dicatat oleh pengguna.

Aplikasi mendukung dua jenis pencatatan tugas, yaitu tugas yang dikaitkan dengan mata kuliah tertentu dan tugas umum tanpa kaitan langsung. Setiap tugas dapat diberi status dan prioritas. Walaupun nilai status dan prioritas disimpan dalam bentuk angka di basis data, aplikasi menampilkannya sebagai label yang lebih mudah dipahami, seperti “Not Started”, “In Progress”, atau “Completed” untuk status, serta “Low”, “High”, atau “Critical” untuk prioritas. Penggunaan label tersebut membantu pengguna memahami konteks tugas dengan cepat tanpa harus melihat representasi angka mentah.

3.2 Use Case

Dalam aplikasi CourseTask, terdapat dua use case utama yang mewakili kebutuhan mahasiswa dalam mengelola aktivitas perkuliahan.

1) Mengelola tugas

Mahasiswa dapat menambah, mengubah, dan menghapus tugas. Setiap tugas dapat dilengkapi dengan judul, deskripsi opsional, tanggal mulai, batas waktu, tingkat prioritas, serta status pengerjaan. Pengguna juga dapat memilih apakah tugas tersebut terkait dengan salah satu mata kuliah yang telah mereka catat sebelumnya.

2) Mengelola mata kuliah

Mahasiswa dapat menambah data mata kuliah mencakup nama mata kuliah, nama pengajar, jadwal, dan deskripsi singkat. Pengguna dapat mengubah maupun menghapus data mata kuliah sesuai kebutuhan. Jika sebuah mata kuliah memiliki tugas terkait, aplikasi menampilkan daftar tugas tersebut di halaman



detail mata kuliah.

Dua use case ini cukup untuk menggambarkan ruang lingkup aplikasi, karena fokus utamanya adalah menyediakan pencatatan tugas yang ringkas dan penyimpanan informasi mata kuliah yang relevan bagi mahasiswa.

3.3 Perancangan Basis Data

Basis data aplikasi dirancang menggunakan dua tabel utama, yaitu lectures dan assignments. Struktur tabel ini sederhana, tetapi sudah memenuhi kebutuhan untuk menyimpan data mata kuliah dan tugas secara terpisah serta mengelola hubungan di antara keduanya.

Tabel lectures menyimpan informasi dasar mata kuliah seperti nama, pengajar, jadwal, dan deskripsi. Sementara tabel assignments menyimpan data tugas yang meliputi judul, deskripsi, prioritas, status, tanggal mulai, dan batas waktu. Kolom lecture_id pada tabel assignments bersifat opsional, memungkinkan pengguna mencatat tugas yang tidak terkait dengan mata kuliah tertentu.

Representasi tabel relasional adalah sebagai berikut:

Tabel 1. Tabel Lectures

Kolom	Tipe Data	Keterangan
id	integer	Primary key, auto-increment
title	varchar	Nama mata kuliah, wajib diisi
lecturer_name	varchar	Nama pengajar, opsional
schedule	varchar	Jadwal kuliah, opsional
description	text	Deskripsi mata kuliah, opsional

Tabel 2. Tabel Assignments

Kolom	Tipe Data	Keterangan
-------	-----------	------------



id	integer	Primary key, auto-increment
lecture_id	integer	Foreign key, boleh NULL (tugas umum tanpa mata kuliah tertentu)
title	Judul tugas	Judul tugas, wajib diisi
description	text	Deskripsi tugas, opsional
priority	integer	Tingkat prioritas tugas, ditampilkan sebagai label dalam aplikasi
status	integer	Status pengerjaan, ditampilkan sebagai label dalam aplikasi
start_date	varchar	Tanggal mulai tugas (yyyy-mm-dd)
deadline_date	varchar	Tanggal tenggat tugas (yyyy-mm-dd)

- Hubungan

Satu mata kuliah dapat memiliki banyak tugas terkait. Namun, tugas dapat pula berdiri sendiri tanpa dikaitkan dengan mata kuliah.

3.4 Implementasi Aplikasi

Pengembangan aplikasi CourseTask dilakukan menggunakan Flutter sebagai kerangka kerja utama. Aplikasi dibangun dengan pendekatan modular, di mana fitur-fitur utama dipisahkan dalam folder sesuai kelompok fungsinya. Hal ini memudahkan proses pengembangan dan meminimalkan risiko inkonsistensi dalam struktur kode.

Untuk pengelolaan data lokal, digunakan Drift sebagai solusi basis data berbasis SQLite. Drift dipilih karena memberikan akses SQL yang lebih terstruktur dan aman dibandingkan melakukan kueri mentah. Setiap tabel diwakili oleh kelas data, dan Drift secara otomatis menangani pembuatan objek, validasi tipe data, serta operasi CRUD.

Manajemen status aplikasi menggunakan Riverpod karena sifatnya yang ringan, sederhana, serta mendukung arsitektur yang bersih. Semua halaman aplikasi mengakses data menggunakan provider yang terhubung langsung ke repository, sehingga perubahan data seperti penambahan atau penghapusan tugas langsung tercermin pada antarmuka pengguna.

Navigasi aplikasi dikelola menggunakan go_router. Pendekatan ini memberikan struktur navigasi yang lebih jelas dan mengurangi kompleksitas saat aplikasi berkembang. Misalnya, ketika pengguna memilih salah satu mata kuliah, aplikasi akan menampilkan halaman detail yang berisi informasi mata kuliah tersebut beserta daftar tugas terkait. Halaman detail ini juga memungkinkan pengguna untuk mengubah atau menghapus data.



Dalam hal antarmuka, desain aplikasi dibuat sederhana dan bersih untuk memprioritaskan keterbacaan. Informasi penting seperti deadline tugas, nama mata kuliah, dan status pengerjaan disajikan dalam kartu-kartu terstruktur sehingga pengguna dapat memahami konteks hanya dengan melihat sekilas tampilan. Label warna pada status dan prioritas membantu pengguna mengenali tingkat urgensi atau progres tugas tanpa harus membuka detail tugas tersebut.

3.5 Pengujian Fungsional

Pengujian dilakukan untuk memastikan aplikasi berjalan stabil dan setiap fitur berfungsi sebagaimana mestinya. Pengujian meliputi:

1) Penambahan data

Pengguna dapat menambah mata kuliah maupun tugas baru. Data tersimpan dengan benar dan muncul pada tampilan utama sesuai kategorinya.

2) Pengubahan data

Form yang tersedia memungkinkan pengguna memperbarui informasi. Perubahan yang disimpan langsung menggantikan data lama tanpa kendala.

3) Penghapusan data

Data mata kuliah atau tugas dapat dihapus. Jika sebuah mata kuliah dihapus, tugas yang terhubung dengannya tetap disimpan sebagai tugas umum.

4) Relasi tugas-mata kuliah

Jika sebuah tugas dikaitkan dengan mata kuliah tertentu, tugas tersebut akan muncul pada halaman detail mata kuliah yang bersangkutan.

5) Navigasi antar halaman

Navigasi melalui bottom navigation bar berfungsi tanpa error, dan transisi antar halaman berjalan lancar.

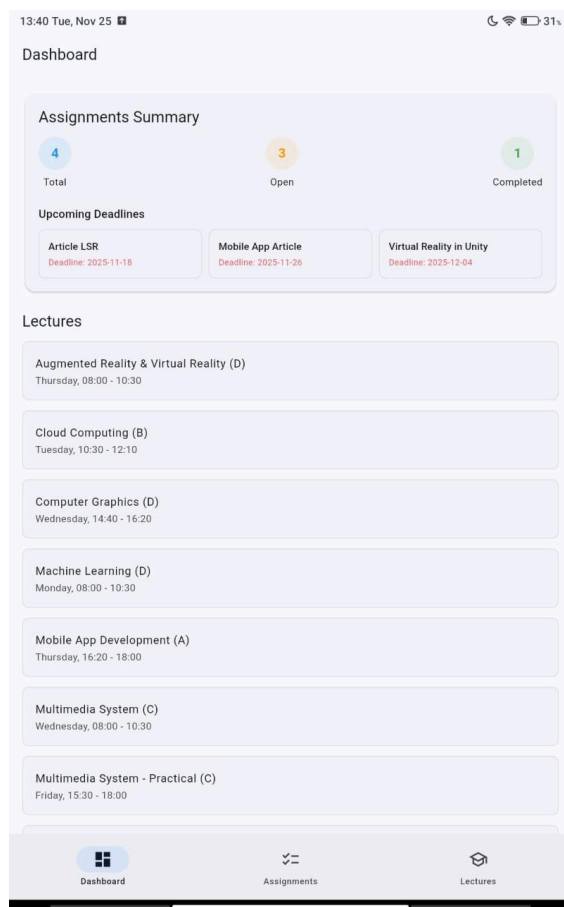
Secara keseluruhan, hasil pengujian menunjukkan bahwa aplikasi dapat digunakan secara normal untuk mencatat dan mengelola aktivitas perkuliahan.

3.6 Tampilan Antarmuka

Tampilan antarmuka CourseTask dirancang dengan pendekatan sederhana dan fokus pada keterbacaan informasi. Setiap halaman menampilkan elemen yang relevan

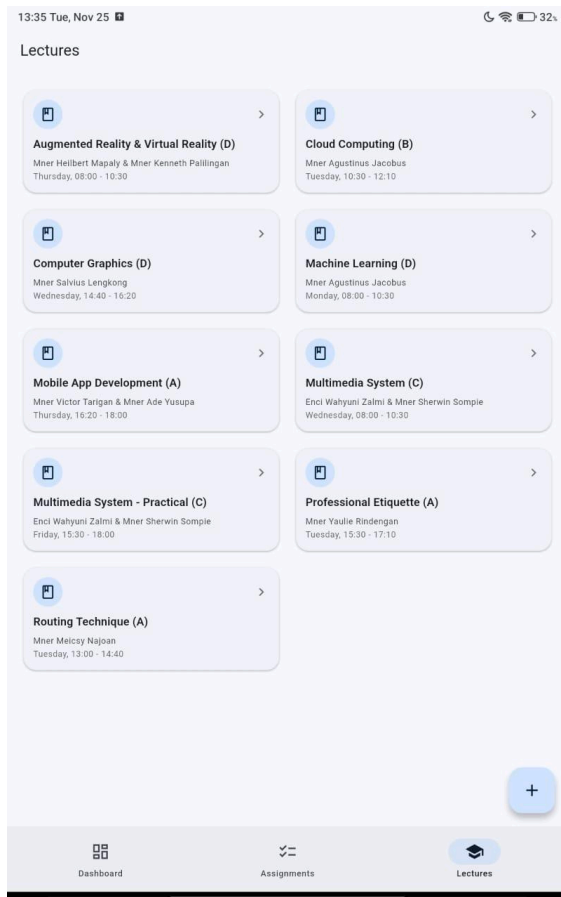
tanpa gangguan visual yang tidak perlu. Bagian ini menampilkan cuplikan antarmuka utama yang merepresentasikan hasil pengembangan aplikasi.

Gambar 1 menunjukkan tampilan Dashboard, yang berisi ringkasan tugas serta daftar mata kuliah yang sudah dicatat oleh pengguna. Bagian ini memudahkan mahasiswa melihat jumlah tugas terbuka, tugas selesai, serta deadline yang akan datang.



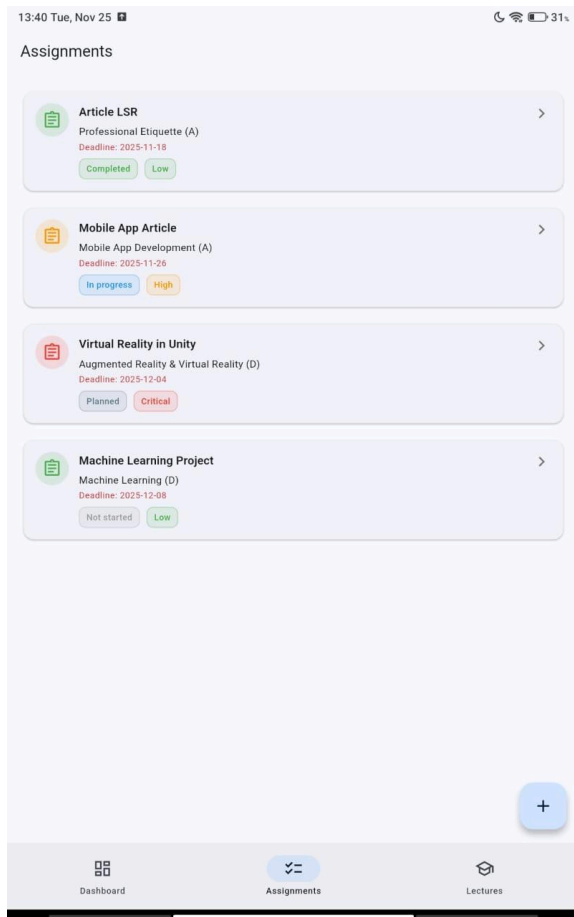
Gambar 1. Tampilan Dashboard yang menampilkan ringkasan tugas dan daftar mata kuliah.

Gambar 2 menampilkan halaman Lectures, yaitu daftar mata kuliah yang ditampilkan dalam bentuk kartu dua kolom. Setiap kartu berisi nama mata kuliah, nama pengajar, serta jadwal kuliah yang memudahkan pengguna mengidentifikasi informasi penting secara cepat.



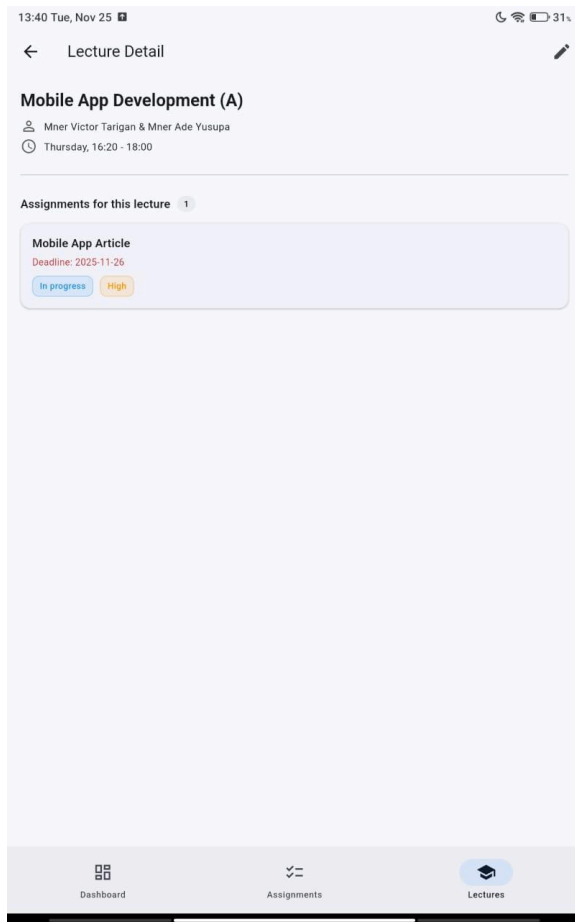
Gambar 2. Halaman Lectures yang menampilkan daftar mata kuliah dalam bentuk kartu dua kolom.

Gambar 3 memperlihatkan halaman Assignments, yang menampilkan daftar tugas beserta label status dan prioritas. Meskipun nilai status dan prioritas disimpan dalam bentuk angka pada basis data, antarmuka menampilkannya sebagai label seperti “Low”, “High”, “Critical”, atau “Completed” sehingga lebih mudah dipahami pengguna.



Gambar 3. Halaman Assignments yang berisi daftar tugas beserta label status dan prioritas.

Gambar 4 memperlihatkan tampilan Detail Mata Kuliah, yang berisi informasi lengkap tentang mata kuliah beserta tugas-tugas yang terkait dengan mata kuliah tersebut. Tampilan ini membantu pengguna memahami konteks dan keterkaitan antara tugas dan mata kuliah.



Gambar 4. Halaman Detail Mata Kuliah yang menampilkan informasi mata kuliah dan daftar tugas terkait.

KESIMPULAN

Penelitian ini menghasilkan aplikasi CourseTask sebagai solusi mobile yang membantu mahasiswa mengelola tugas dan informasi mata kuliah secara lebih teratur. Aplikasi ini dikembangkan menggunakan pendekatan Waterfall dengan tahapan yang terstruktur, mulai dari analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian. Hasil pengembangan menunjukkan bahwa fitur utama yang dirancang—seperti pencatatan mata kuliah, pencatatan tugas, serta keterkaitan antara keduanya—dapat berfungsi dengan baik dan memberikan manfaat nyata bagi pengguna.

Penyimpanan data menggunakan basis data lokal memungkinkan aplikasi digunakan tanpa koneksi internet, sehingga tetap relevan bagi mahasiswa yang membutuhkan pencatatan cepat dan konsisten dalam berbagai kondisi. Antarmuka yang sederhana dan fokus pada keterbacaan memudahkan pengguna untuk memahami informasi penting seperti deadline, status pengerjaan, serta tingkat prioritas tugas.



Secara keseluruhan, CourseTask mampu memenuhi tujuan pengembangannya sebagai aplikasi yang praktis, ringkas, dan mudah digunakan. Aplikasi ini dapat menjadi alternatif yang efektif bagi mahasiswa yang membutuhkan alat bantu untuk mengatur aktivitas perkuliahan sehari-hari. Ke depan, aplikasi ini masih dapat dikembangkan lebih lanjut, misalnya dengan penambahan fitur pengingat atau sinkronisasi lintas perangkat apabila dibutuhkan.

DAFTAR PUSTAKA

- [1] Google. Flutter — Build Apps for Any Screen. 2025.
<https://flutter.dev>
- [2] Google. Flutter Architectural Overview. 2025.
<https://docs.flutter.dev/resources/architectural-overview>
- [3] J. A. Puryanto & H. Akbar, “Performance Analysis of Provider and Riverpod State Management Library on Flutter Applications,” *Journal of Technology and Informatics (JoTI)*, vol. 7, no. 2, 2025.
<https://doi.org/10.37802/joti.v7i2.1164>
- [4] A. Donvir, A. Jain, & P. Saraswathi, “Application State Management in Modern Web and Mobile Applications: A Comprehensive Review,” *arXiv preprint*, 2024.
<https://arxiv.org/abs/2407.19318>
- [5] S. P. Tiwari, “Knowledge Enhancement and Mobile Technology: Improving Effectiveness and Efficiency,” *arXiv preprint*, 2022.
<https://arxiv.org/abs/2208.04706>
- [6] F. A. E. Atmojo & I. M. Widiartha, “Perbandingan Kinerja Local Database Pada Aplikasi Mobile Dengan Flutter,” *JNATIA*, vol. 1, 2022.
<https://jurnal.harianregional.com/jnatia/id-92578>
- [7] Drift Contributors. Drift — Reactive Persistence Library for Flutter & Dart.
<https://pub.dev/packages/drift>
- [8] H. Bomo, “SQLite Database Using Drift — How to Develop a Feature in Flutter Project,” *Medium*, 2023.
<https://medium.com/@huguesarnold/sqlite-database-using-drift-how-to-develop-a-feature-in-flutter-project-part-5-96d142a38bda>
- [9] D. Slepnev, “State Management Approaches in Flutter,” *Bachelor’s Thesis*, Helsinki Metropolia University of Applied Sciences, 2020.



https://www.theseus.fi/bitstream/handle/10024/355086/Dmitrii_Slepnev.pdf

[10] Google. Persist Data with SQLite - Flutter Cookbook.

<https://docs.flutter.dev/cookbook/persistence/sqlite>

[11] R. S. Pressman & B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed., 2020.

<https://www.mheducation.com/highered/product/software-engineering-practitioner-s-approach-pressman-maxim/M9781259872976.html>