
Software Requirements Specification

for

Kuliah.ku

Version 1.0 approved

Prepared by :

Kevin Ardi Setyawan 2313025057

Kelas A

27 September 2025

Table of Contents

1. Pendahuluan	1
1.1 Latar Belakang	1
1.2 Lingkup Masalah	1
2. Deskripsi Umum Perangkat Lunak	3
2.1 Deskripsi Umum Sistem	3
2.3 Batasan	3
3. Deskripsi Kebutuhan	5
3.1 Kebutuhan Antarmuka	5
3.2 Kebutuhan Fungsional	6
3.3 Kebutuhan Non Fungsional	9
3.4 Model Analisis	10
3.4.1 Diagram Use Case	10
3.4.1.1 Diagram Use Case 1	10
3.4.2 Diagram Kelas	11
3.4.2.1 Diagram Kelas 1	11
3.4.2.2 Diagram Kelas 2	12
3.4.3 Diagram Data Flow (DFD)	13
3.4.3.1 Context Diagram	13
3.4.3.2 DFD 1	14
3.4.4 Activity Diagram	15
3.4.5 Entity Relationship Diagram (ERD)	16
3.4.5.1 ERD 1	16
3.4.5.2 ERD 2	17
3.5 Software Development Life Cycle (SDLC)	18
4. Lain-Lain	20
4.1 Github	20

Revision History

Name	Date	Reason For Changes	Version

1. Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi mobile memberikan peluang baru di dunia pendidikan, khususnya dalam mendukung aktivitas belajar mahasiswa. Tapi, masih banyak mahasiswa yang kesulitan dalam mengatur jadwal kuliah, mengelola tugas, dan mengelola keuangan dengan baik. Saya membuat aplikasi bernama Kuliah.ku yang dirancang untuk menjadi solusi digital yang membantu mahasiswa dalam mengatur perkuliahan, manajemen waktu, serta meningkatkan efektivitas di dunia perkuliahan. Dengan fitur seperti jadwal kuliah, manajemen tugas, catatan digital, dan manajemen keuangan, aplikasi ini diharapkan dapat menjadi asisten belajar pribadi mahasiswa.

1.2 Lingkup Masalah

A. Tujuan

Proyek ini bertujuan untuk:

1. Menyediakan satu platform tunggal bagi mahasiswa untuk mengelola semua aspek penting dari kehidupan perkuliahan mereka, yang mencakup jadwal mata kuliah, tenggat waktu tugas dan anggaran keuangan pribadi khusus untuk mahasiswa.
2. Membantu mahasiswa menjadi lebih terorganisir dengan fitur-fitur proaktif seperti pengingat (reminder) untuk kelas yang akan datang, notifikasi tenggat waktu tugas, dan ringkasan keuangan mingguan/bulanan.
3. Mengembangkan aplikasi dengan desain antarmuka (UI) dan pengalaman pengguna (UX) yang modern, sederhana, dan mudah digunakan, sehingga mahasiswa dapat mengadopsi aplikasi ini dalam rutinitas harian mereka tanpa kesulitan.

B. Manfaat

Dengan adanya sistem ini, mahasiswa dapat:

1. Peningkatan Produktivitas dan Manajemen Waktu:
Dengan jadwal dan daftar tugas yang terintegrasi, mahasiswa dapat merencanakan waktu belajar dan kegiatan lainnya dengan lebih efektif, mengurangi risiko lupa akan tugas atau jadwal penting.
2. Organisasi Akademik yang Lebih Baik:
Semua catatan, tugas, dan jadwal tersimpan secara terpusat dan terstruktur per mata kuliah, memudahkan mahasiswa untuk mengakses informasi yang mereka butuhkan saat belajar atau mengerjakan tugas.
3. Mengurangi Tingkat Stres:
Dengan gambaran yang jelas tentang semua tanggung jawab akademik dan tenggat waktu, tingkat kecemasan dan stres yang sering dialami mahasiswa karena merasa "kewalahan" dapat diminimalkan.
4. Peningkatan Kesadaran Finansial:
Fitur “pelacakan keuangan” membantu mahasiswa memahami ke mana uang mereka pergi, mendorong mereka untuk lebih bijak dalam pengeluaran dan menabung.
5. Aksesibilitas dan Portabilitas:
Sebagai aplikasi mobile, "kuliah.ku" memungkinkan mahasiswa untuk mengakses dan mengelola semua informasi penting mereka kapan saja dan di mana saja, hanya dengan menggunakan smartphone mereka.

C. Batasan Proyek Akhir

Dalam pengembangan aplikasi kuliah.ku, terdapat beberapa batasan yang perlu diperhatikan agar implementasi berjalan sesuai rencana.

1. Platform:
Pengembangan awal akan difokuskan pada satu platform mobile saja, yaitu Android. Pengembangan untuk platform lain (seperti iOS) akan dipertimbangkan untuk fase berikutnya.
2. Fungsionalitas Online/Offline:
Aplikasi pada versi awal akan dirancang untuk berfungsi secara offline-first. Semua data (jadwal, tugas, catatan, keuangan) akan disimpan secara lokal di perangkat pengguna. Fitur sinkronisasi cloud atau backup online tidak akan menjadi bagian dari rilis pertama.

3. Bukan Aplikasi Kolaborasi:
Aplikasi ini dirancang untuk penggunaan individu. Tidak akan ada fitur untuk berbagi jadwal, tugas, atau catatan dengan pengguna lain (fitur kerja kelompok) pada versi ini.
4. Batasan Fitur Keuangan:
Fitur keuangan terbatas pada pencatatan pemasukan dan pengeluaran manual khusus untuk kehidupan sebagai mahasiswa. Aplikasi tidak akan terintegrasi dengan rekening bank, e-wallet, atau layanan finansial pihak ketiga lainnya.
5. Batasan Fitur Catatan:
Fitur catatan akan mendukung input teks dan checklist sederhana. Fitur yang lebih canggih seperti menyisipkan gambar, rekaman suara, atau lampiran file tidak akan disertakan pada versi awal.
6. Bahasa:
Aplikasi dan seluruh konten di dalamnya akan disajikan dalam Bahasa Indonesia.

2. Deskripsi Umum Aplikasi

2.1 Deskripsi Umum Sistem

Aplikasi "**kuliah.ku**" adalah sebuah sistem mandiri yang beroperasi di perangkat mobile berbasis Android. Aplikasi ini dirancang sebagai asisten pribadi digital yang komprehensif, bertujuan untuk membantu mahasiswa mengelola dan mengorganisir empat pilar utama dalam kehidupan akademik mereka: **Jadwal, Tugas, Catatan, dan Keuangan**.

Secara fungsional, sistem ini terdiri dari empat modul utama yang terintegrasi:

1. Modul Manajemen Jadwal:

- Pengguna dapat memasukkan, menyunting, dan menghapus jadwal mata kuliah, praktikum, atau kegiatan akademik lainnya.
- Setiap entri jadwal akan berisi informasi detail seperti nama mata kuliah, nama dosen, dan waktu (hari dan jam).
- Sistem akan menampilkan jadwal dalam format harian atau mingguan untuk kemudahan visualisasi.
- Akan ada fitur pengingat (*reminder*) opsional untuk memberitahu pengguna beberapa saat sebelum kelas dimulai.

2. Modul To Do List:

- Pengguna dapat membuat daftar pekerjaan dalam berbagai kategori seperti "Kuliah", "Pribadi", atau "Wishlist".
- Sistem memungkinkan pengguna untuk menambahkan item, menandainya sebagai selesai, menyaring berdasarkan kategori, dan mengatur tenggat waktu (khususnya untuk tugas kuliah).
- Modul ini berfungsi sebagai pusat untuk semua hal yang perlu diingat atau dikerjakan, baik itu tugas akademis, rencana pribadi, maupun daftar keinginan.

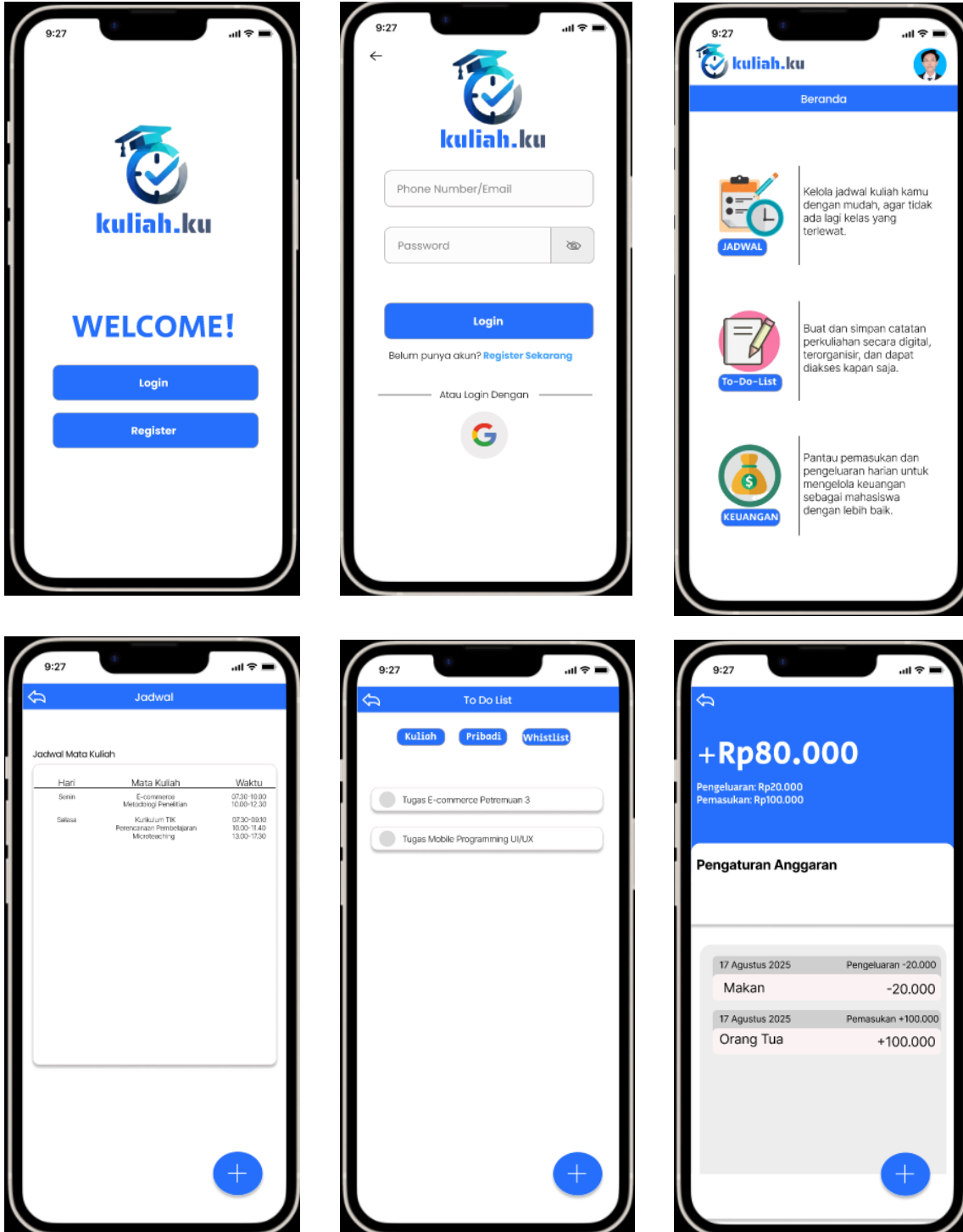
3. Modul Manajemen Keuangan:

- Pengguna dapat secara manual mencatat setiap transaksi pemasukan (misal: uang saku bulanan) dan pengeluaran (misal: biaya makan, fotokopi, buku).
- Setiap transaksi akan memiliki tanggal dan deskripsi singkat.
- Sistem akan secara otomatis menghitung dan menampilkan saldo sisa.
- Akan tersedia ringkasan (dashboard) sederhana yang menunjukkan total pemasukan dan pengeluaran dalam periode waktu tertentu (misalnya, bulanan).

Keempat modul ini dirancang untuk bekerja secara harmonis dalam satu aplikasi, menghilangkan kebutuhan mahasiswa untuk menggunakan beberapa aplikasi terpisah untuk setiap fungsi tersebut.

3. Deskripsi Kebutuhan

3.1 Kebutuhan Antarmuka (UI/UX)



3.2 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan fungsionalitas atau layanan spesifik yang harus disediakan oleh sistem. Setiap kebutuhan diberi ID unik untuk kemudahan pelacakan (contoh: F-JDL-01).

3.2.1 Fungsionalitas Modul Beranda

- **F-BRD-01 (Beranda Utama):** Sistem harus menampilkan halaman Beranda sebagai layar utama setelah aplikasi dibuka. Halaman ini harus berisi tiga menu navigasi utama yang jelas: Jadwal, To-Do-List, dan Keuangan.

3.2.2 Fungsionalitas Modul Manajemen Jadwal

- **F-JDL-01:**
Sistem harus memungkinkan pengguna untuk **menambahkan entri jadwal baru** dengan mengisi kolom: Nama Mata Kuliah, Hari, Waktu, dan Nama Dosen (opsional).
- **F-JDL-02:**
Sistem harus menampilkan jadwal kuliah dalam format daftar yang dikelompokkan berdasarkan hari (contoh: Selasa).
- **F-JDL-03:**
Pengguna harus dapat **menyunting (edit)** setiap detail dari entri jadwal yang sudah ada.
- **F-JDL-04:**
Pengguna harus dapat **menghapus** entri jadwal yang sudah tidak relevan.
- **F-JDL-05:**
Saat pengguna mengetuk sebuah jadwal, sistem harus menampilkan **jendela detail** yang berisi semua informasi terkait jadwal tersebut.

3.2.3 Fungsionalitas Modul To Do List

- **F-TDL-01 (Menambah Item To-Do):**
Sistem harus memungkinkan pengguna untuk **menambahkan item baru** ke dalam To Do List melalui tombol tambah (+). Saat menambahkan, pengguna harus dapat mengisi:
 - 1) Nama/Judul Tugas.
 - 2) **Kategori** (pilihan: Kuliah, Pribadi, Wishlist).
 - 3) Tanggal & Jam Tenggat Waktu (khususnya untuk kategori "Kuliah").
 - 4) Deskripsi (opsional).
- **F-TDL-02 (Melihat Daftar To-Do):**

Sistem harus **menampilkan semua item To Do List** yang belum selesai. Secara default, sistem dapat menampilkan semua kategori atau kategori "Kuliah".

- **F-TDL-03 (Menyaring Daftar To-Do):**
Sesuai dengan desain UI, sistem harus memungkinkan pengguna untuk **menyaring (filter) daftar To Do** berdasarkan kategori yang dipilih dengan menekan tombol "Kuliah", "Pribadi", atau "Wishlist".
- **F-TDL-04 (Menandai Selesai):**
Pengguna harus dapat **menandai sebuah item sebagai "selesai"** dengan menekan *checkbox* (lingkaran) di sebelah kiri nama item. Setelah ditandai, item tersebut harus memberikan umpan balik visual (misalnya, teks dicoret atau warna berubah menjadi abu-abu).
- **F-TDL-05 (Mengubah Item To-Do):** Pengguna harus dapat **menyunting** detail dari item To Do yang sudah ada (misalnya, mengubah nama, tenggat waktu, atau kategori). Aksi ini dapat diakses dengan menekan item tersebut.
- **F-TDL-06 (Menghapus Item To-Do):** Pengguna harus dapat **menghapus** sebuah item dari To Do List.

3.2.3 Fungsionalitas Modul Manajemen Keuangan

- **F-KEU-01:**
Pengguna harus dapat mencatat transaksi pemasukan dengan mengisi kolom: Jumlah Uang, Tanggal, dan Keterangan.
- **F-KEU-02:**
Pengguna harus dapat mencatat transaksi pengeluaran dengan mengisi kolom: Jumlah Uang, Tanggal, dan Keterangan.
- **F-KEU-03:**
Di bawah kartu ringkasan, sistem harus menampilkan **daftar riwayat transaksi** yang diurutkan secara kronologis, di mana setiap item menunjukkan keterangan, tanggal, dan jumlahnya (dengan tanda + untuk pemasukan dan - untuk pengeluaran).
- **F-KEU-04:**
Sistem harus menampilkan **kartu ringkasan (summary card)** di bagian atas halaman Keuangan yang berisi informasi: **Total Pemasukan, Total Pengeluaran,** dan **Saldo Sisa** untuk periode berjalan.
- **F-KEU-05:**
Pengguna harus dapat menghapus entri transaksi jika terjadi kesalahan input.

3.2.4 Modul Notifikasi

- **F-NOT-01:**
Sistem harus dapat mengirimkan notifikasi pengingat 15 menit sebelum jadwal kuliah dimulai.
- **F-NOT-02:**
Sistem harus dapat mengirimkan notifikasi pengingat 24 jam sebelum tenggat waktu sebuah item To Do yang berada dalam kategori 'Kuliah'.

3.2.5 Fungsionalitas Autentikasi

- **F-AUT-01:** Sistem harus menyediakan fitur registrasi pengguna baru.
- **F-AUT-02:** Sistem harus menyediakan fitur login untuk pengguna terdaftar.
- **F-AUT-03:** Sistem harus menjaga sesi pengguna tetap aktif (tetap login) setelah menutup aplikasi.

3.3 Kebutuhan Non Fungsional

Ini adalah daftar standar kualitas, batasan teknis, dan cara sistem seharusnya bekerja.

3.3.1 Kinerja (Performance)

- **NF-KIN-01:** Waktu muat aplikasi dari saat ikon diklik hingga layar utama siap digunakan tidak boleh lebih dari 3 detik.
- **NF-KIN-02:** Respon antarmuka terhadap input pengguna (ketukan, gesekan) harus terasa instan, dengan latensi di bawah 200 milidetik.
- **NF-KIN-03:** Aplikasi harus dapat berjalan lancar pada perangkat dengan spesifikasi menengah (RAM 2GB).

3.3.2 Keamanan (Security)

- **NF-SEC-01:** Semua data pengguna harus disimpan secara lokal di dalam *private storage* aplikasi, sehingga tidak dapat diakses oleh aplikasi lain.
- **NF-SEC-02:** Aplikasi tidak boleh meminta izin (permission) Android yang tidak relevan dengan fungsionalitas inti (misalnya, izin kontak atau lokasi).

3.3.2 Keandalan (Reliability)

- **NF-REL-01:** Data yang sudah disimpan oleh pengguna tidak boleh hilang atau rusak jika aplikasi ditutup secara paksa atau perangkat di-restart.
- **NF-REL-02:** Aplikasi harus memiliki tingkat stabilitas tinggi, dengan target *crash-free users* di atas 99%.

3.3.3 Usability (Kemudahan Penggunaan)

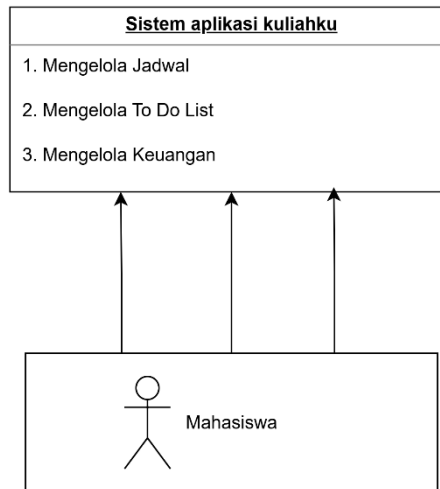
- **NF-USA-01:** Antarmuka aplikasi harus konsisten di semua layar, menggunakan palet warna, jenis huruf, dan gaya ikon yang seragam.
- **NF-USA-02:** Pengguna baru harus dapat memahami dan menggunakan fungsi-fungsi utama dalam waktu kurang dari 5 menit tanpa memerlukan manual atau tutorial.
- **NF-USA-03:** Semua teks, label, dan instruksi dalam aplikasi harus menggunakan **Bahasa Indonesia** yang baku dan mudah dipahami.

3.3.4 Kompatibilitas (Compatibility)

- **NF-COM-01:** Aplikasi harus dapat diinstal dan berjalan dengan baik pada sistem operasi **Android versi 8.0 (Oreo)** ke atas.

3.4 Model Analisis

3.4.1 Diagram Use Case

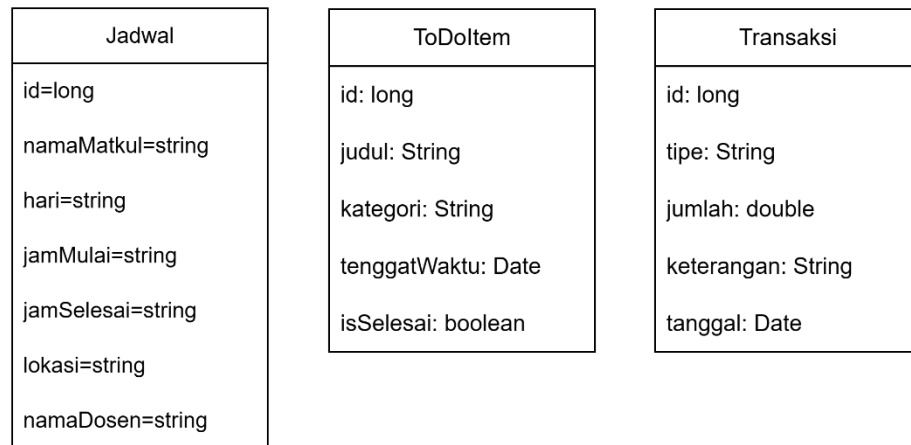


Penjelasan:

- Aktor **Mahasiswa** terhubung (dapat menggunakan) tiga fungsi utama (use case) di dalam sistem: **Mengelola Jadwal**, **Mengelola To Do List**, dan **Mengelola Keuangan**.
- Deskripsi:
 1. **Aktor:** Mahasiswa
 2. **Use Cases:**
 - 1) **Mengelola Jadwal**
 - Menambah Jadwal
 - Melihat Jadwal
 - Mengubah Jadwal
 - Menghapus Jadwal
 - 2) **Mengelola To Do List**
 - Menambah Item To Do
 - Melihat & Menyaring To Do
 - Menandai Selesai
 - Menghapus Item To Do
 - 3) **Mengelola Keuangan**
 - Mencatat Pemasukan
 - Mencatat Pengeluaran
 - Melihat Riwayat Transaksi
 - Melihat Ringkasan Keuangan

3.4.2 Diagram Kelas

3.4.2.1 Diagram Kelas 1



Penjelasan :

1. Kelas Jadwal

Kelas ini merepresentasikan satu entri jadwal mata kuliah.

- **id: long:** Nomor unik untuk setiap jadwal, biasanya dibuat otomatis oleh database (*primary key*).
- **namaMatkul: String:** Nama mata kuliah, contoh: "Pemrograman Mobile".
- **hari: String:** Hari pelaksanaan kuliah, contoh: "Senin".
- **jamMulai: String:** Waktu mulai kuliah, contoh: "13:30".
- **jamSelesai: String:** Waktu selesai kuliah, contoh: "15:10".
- **lokasi: String:** Ruang atau gedung tempat kuliah, contoh: "Lab RPL".
- **namaDosen: String:** Nama dosen pengampu (opsional).

2. Kelas ToDoItem

Kelas ini merepresentasikan satu item pekerjaan dalam To Do List.

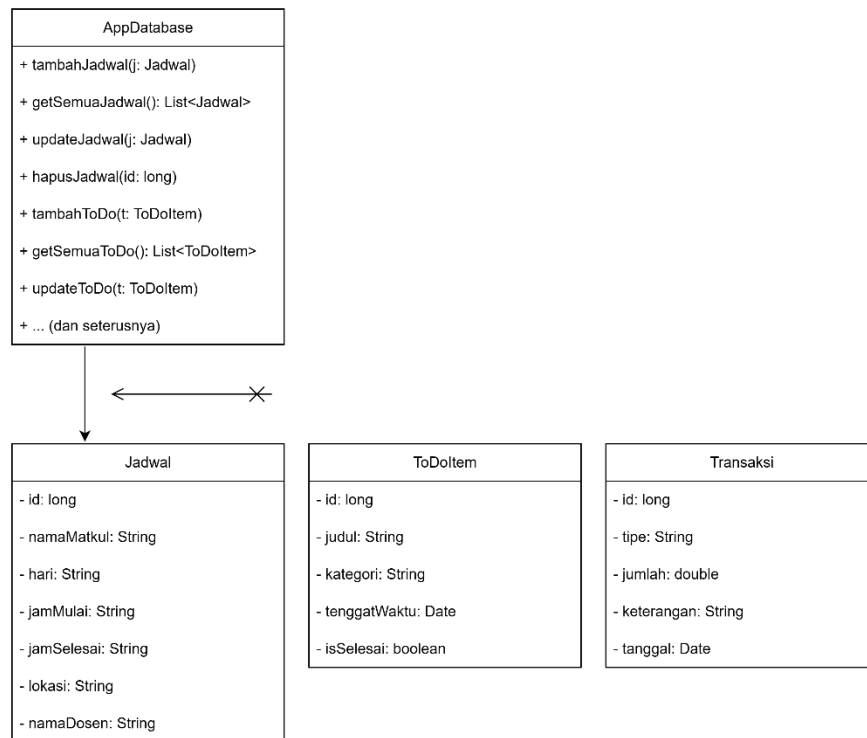
- **id: long:** Nomor unik untuk setiap item.
- **judul: String:** Nama atau judul dari pekerjaan, contoh: "Mengerjakan UI/UX Tugas 3".
- **kategori: String:** Kategori dari item tersebut, yaitu "Kuliah", "Pribadi", atau "Wishlist".
- **tenggatWaktu: Date:** Tanggal dan waktu tenggat (opsional, tapi penting untuk kategori Kuliah).
- **isSelesai: boolean:** Status pekerjaan, true jika sudah selesai dan false jika belum.

3. Kelas Transaksi

Kelas ini merepresentasikan satu catatan transaksi keuangan.

- **id: long:** Nomor unik untuk setiap transaksi.
- **tipe: String:** Jenis transaksi, yaitu "Pemasukan" atau "Pengeluaran".
- **jumlah: double:** Besarnya nominal uang dalam transaksi.
- **keterangan: String:** Deskripsi singkat transaksi, contoh: "Beli buku" atau "Uang saku".
- **tanggal: Date:** Tanggal terjadinya transaksi.

3.4.2.2 Diagram Kelas 2



Penjelasan:

1. **Detail Atribut:** Setiap atribut kini memiliki **tipe data yang spesifik** (contoh: id: long, isSelesai: boolean). Ini adalah informasi yang dibutuhkan saat membuat tabel database atau kelas di kode Anda.
2. **Kelas AppDatabase:** Ini adalah kelas sentral yang sangat penting.
 - Ia bertindak sebagai **jembatan** antara UI (tampilan aplikasi) dan penyimpanan data lokal Anda (misalnya database SQLite).
 - **Methods (Fungsi):** Kelas ini berisi semua fungsi yang Anda perlukan untuk memanipulasi data, seperti tambahJadwal(), getSemuaToDo(), hapusTransaksi(),

dan lain-lain. Saat Anda ingin menyimpan jadwal baru, Anda akan memanggil method tambahJadwal() dari kelas ini.

3. **Hubungan (<>--*)**: Simbol ini menunjukkan hubungan **Komposisi**, yang artinya AppDatabase "terdiri dari" atau "memiliki" banyak (*) objek Jadwal, ToDoItem, dan Transaksi. Jika AppDatabase dihancurkan, semua datanya juga akan hilang.

3.4.3 Diagram Data Flow (DFD)

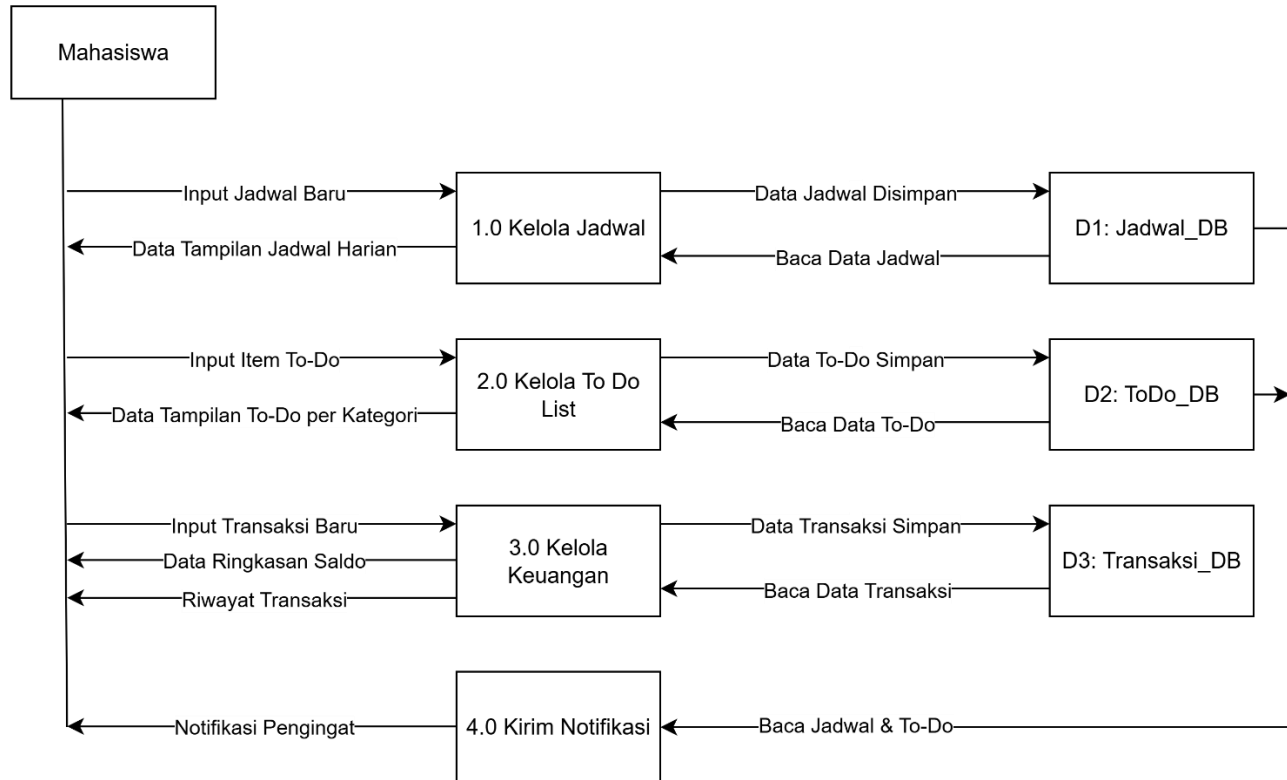
3.4.3.1 Context Diagram



Penjelasan:

- 1) **Mahasiswa** (Entitas Eksternal): Sumber dan tujuan data.
- 2) **Aplikasi kuliah.ku** : Keseluruhan aplikasi Anda.
- 3) **Panah (Aliran Data)**:
 - Mahasiswa memberikan **Input Data** (seperti detail jadwal baru, item to-do, atau data transaksi) ke dalam sistem.
 - Sistem memberikan output berupa **Tampilan & Notifikasi** (seperti menampilkan jadwal, daftar to-do, laporan keuangan, dan pengingat) kepada Mahasiswa

3.4.3.2 DFD 1

**Penjelasan Alur Data**

Berikut rincian bagaimana data bergerak untuk setiap fitur utama sesuai dengan UI Anda:

1. Alur Data Jadwal

- Anda memasukkan detail jadwal baru melalui form (**Input Jadwal Baru**).
- Data ini diterima oleh proses **1.0 Kelola Jadwal**, yang kemudian menyimpannya ke dalam database **D1: Jadwal_DB**.
- Saat Anda membuka halaman Jadwal, proses yang sama membaca data dari **D1: Jadwal_DB** dan mengirimkannya kembali ke antarmuka Anda sebagai **Data Tampilan Jadwal Harian**.

2. Alur Data To-Do List

- Anda memasukkan item to-do baru (Input Item To-Do).
- Proses 2.0 Kelola To Do List menerima dan menyimpan data ini ke D2: ToDo_DB.
- Saat Anda melihat daftar atau memfilternya berdasarkan kategori, proses ini membaca data yang relevan dari D2: ToDo_DB dan menampilkannya sebagai Data Tampilan To-Do per Kategori.

3. Alur Data Keuangan

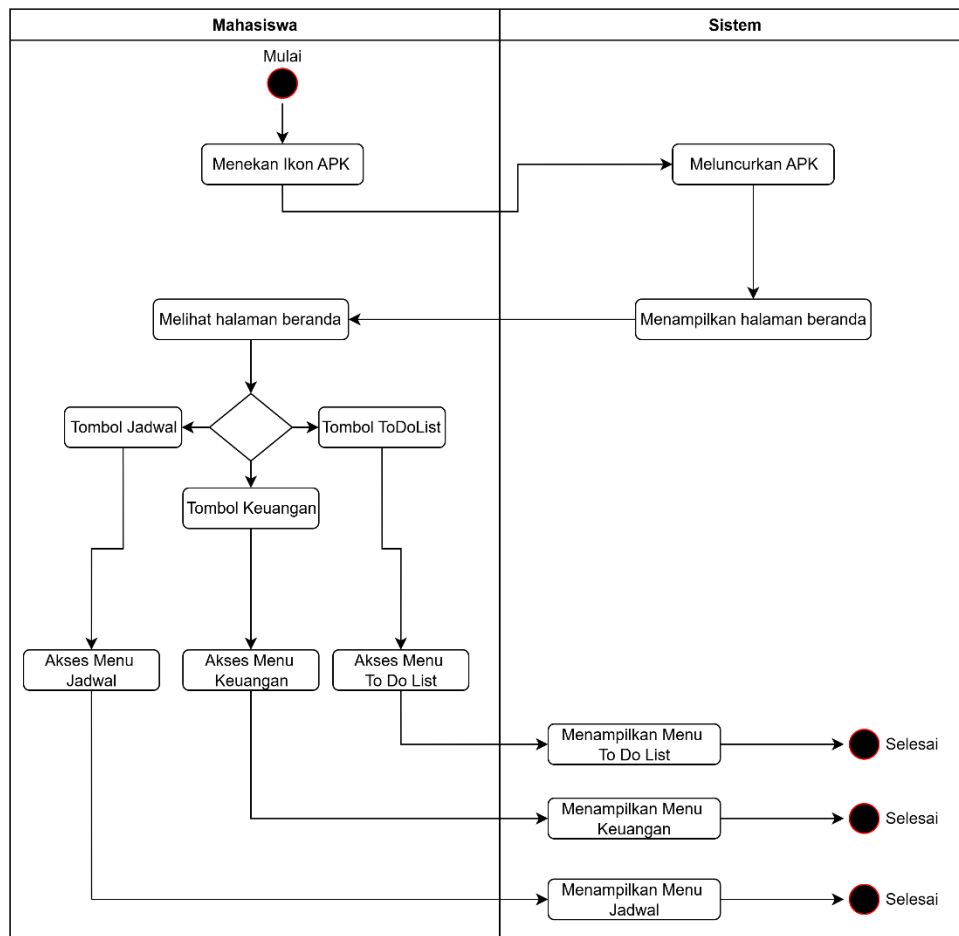
- Anda memasukkan transaksi baru, baik pemasukan maupun pengeluaran (Input Transaksi Baru).
- Proses 3.0 Kelola Keuangan menyimpannya ke D3: Transaksi_DB.
- Untuk menampilkan halaman Keuangan, proses ini membaca data dari D3: Transaksi_DB dan mengirimkannya dalam dua bentuk:

- **Data Ringkasan Saldo:** Data yang sudah dihitung (Total Pemasukan, Pengeluaran, Saldo) untuk ditampilkan di kartu ringkasan atas.
- **Riwayat Transaksi:** Daftar mentah transaksi untuk ditampilkan di bagian bawah.

4. Alur Data Notifikasi

- Proses 4.0 Kirim Notifikasi berjalan di latar belakang.
- Proses ini secara periodik membaca data dari D1: Jadwal_DB dan D2: ToDo_DB untuk memeriksa apakah ada jadwal atau tugas yang akan datang.
- Jika ada, proses ini akan mengirimkan Notifikasi Pengingat ke perangkat Anda.

3.4.4 Activity Diagram



Penjelasan Alur Kerja

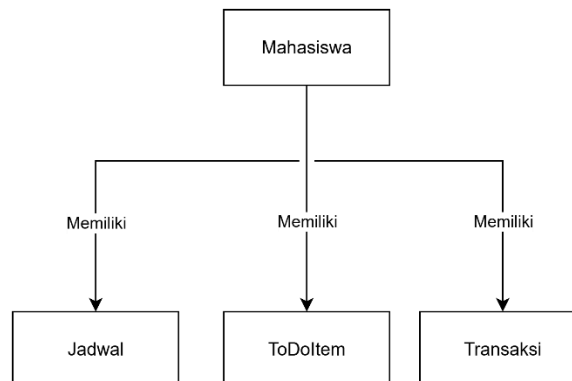
Diagram di atas menunjukkan alur navigasi dasar sebagai berikut:

1. **Mulai** : Proses diawali saat Anda sebagai **Mahasiswa** siap menggunakan aplikasi.
2. **Menekan Ikon Aplikasi**: Anda menekan ikon "kuliah.ku" di ponsel Anda.

3. **Meluncurkan Aplikasi: Sistem** merespons dengan memulai dan memuat aplikasi.
4. **Menampilkan Beranda:** Sistem kemudian menampilkan halaman utama (Beranda atau Dasbor) yang berisi tiga menu utama.
5. **Pilih Menu <◇>:** Ini adalah titik keputusan. Anda dihadapkan pada tiga pilihan menu.
6. **Tiga Kemungkinan Alur:**
 - Jika Anda menekan **Tombol Jadwal**, sistem akan menampilkan **Halaman Jadwal**, dan alur ini selesai.
 - Jika Anda menekan **Tombol To-Do**, sistem akan menampilkan **Halaman To-Do List**, dan alur ini selesai.
 - Jika Anda menekan **Tombol Keuangan**, sistem akan menampilkan **Halaman Keuangan**, dan alur ini selesai.
7. **Selesai :** Proses dianggap selesai setelah Anda berhasil mengakses salah satu dari tiga menu dasar tersebut.

3.4.5 Entity Relationship Diagram (ERD)

3.4.5.1 ERD 1



Penjelasan Hubungan (Kardinalitas):

- Seorang **Mahasiswa** dapat **memiliki BANYAK (<)** **Jadwal**.
- Seorang **Mahasiswa** dapat **memiliki BANYAK (<)** **ToDoItem**.
- Seorang **Mahasiswa** dapat **memiliki BANYAK (<)** **Transaksi**.

Ini adalah hubungan **satu-ke-banyak (one-to-many)**.

3.4.5.2 ERD 2

Jadwal		
Nama Kolom	Tipe Data	Keterangan
id_jadwal	STRING	Primary Key (ID Dokumen unik dari Firestore)
userId	STRING	Foreign Key (FK) 🗝 ke ID Pengguna
nama_matkul	STRING	Nama mata kuliah
hari	STRING	Hari perkuliahan
jam_mulai	STRING	Waktu mulai kuliah (format HH:MM)
jam_selesai	STRING	Waktu selesai kuliah (format HH:MM)
lokasi	STRING	Lokasi atau ruang kelas
nama_dosen	STRING	Nama dosen pengampu (opsional)

ToDo_Item		
Nama Kolom	Tipe Data	Keterangan
id_todo	String	Primary Key (PK)
userId	String	Foreign Key ke ID pengguna dari Authentication
judul	String	judul atau nama tugas
kategori	String	"Kuliah", "Pribadi", atau "Wishlist"
tenggat_waktu	Timestamp	Tipe data tanggal/waktu di Firebase
status_selesai	BOOLEAN	true jika selesai, false jika belum

Transaksi		
Nama Kolom	Tipe Data	Keterangan
id_transaksi	String	Primary Key (PK)
userId	String	Foreign Key ke ID pengguna dari Authentication
tipe	String	"Pemasukan" / "Pengeluaran"
jumlah	NUMBER	Nominal uang
keterangan	STRING	deskripsi singkat transaksi
tanggal	TIMESTAMP	Tipe data tanggal/waktu standar Firebase

3.5 Software Development Life Cycle (SDLC)

Model SDLC: Agile dengan Pendekatan Prototyping

Metode ini menggabungkan fleksibilitas dari Agile dengan fokus pada visualisasi dan umpan balik dari Prototyping. Ini adalah pendekatan modern yang sangat umum digunakan dalam pengembangan aplikasi mobile saat ini.

Keunggulan Memakai Metode ini:

1. **Fleksibilitas Tinggi:**

Agile memungkinkan saya untuk beradaptasi dengan perubahan. Contoh terbaiknya adalah keputusan saya untuk beralih dari database lokal ke Firebase di tengah-tengah proses perancangan. Model yang kaku seperti Waterfall akan membuat perubahan seperti ini sangat sulit.

2. **Pengembangan Bertahap (Iteratif):**

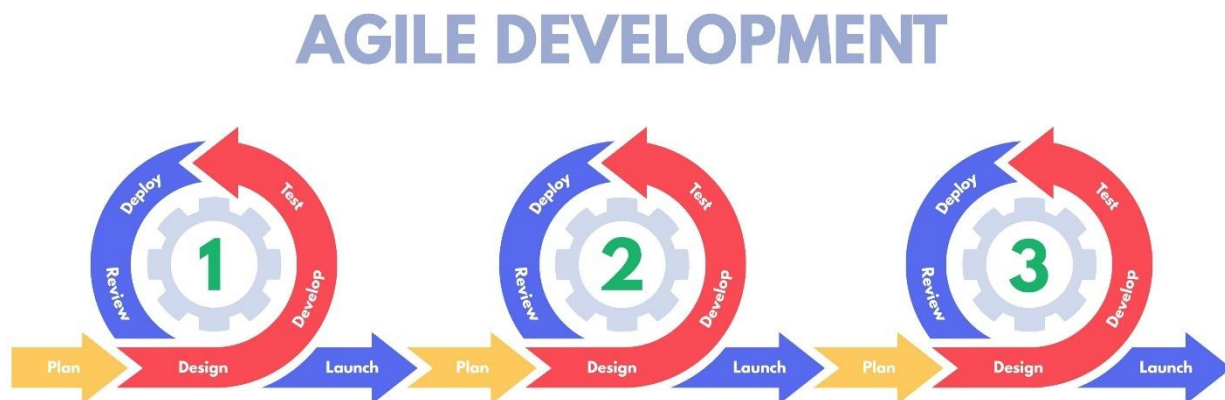
Saya tidak perlu membangun semuanya sekaligus. Dengan Agile, saya bisa memecah pekerjaan menjadi bagian-bagian kecil yang fungsional (disebut *sprint* atau iterasi). Saya bisa fokus membangun fitur **Jadwal** sampai selesai dan berfungsi, lalu pindah ke fitur **To-Do List**, dan seterusnya. Ini membuat proyek lebih mudah dikelola.

3. **Fokus pada Prototipe:**

Saya secara alami sudah menerapkan pendekatan ini dengan membuat desain UI/UX terlebih dahulu. Dengan melihat prototipe visual, kita bisa langsung memahami apa yang akan dibuat, mengurangi kesalahpahaman, dan memastikan produk akhir sesuai dengan visi saya.

4. **Mengurangi Risiko Kegagalan:**

Karena saya membangun dan menguji aplikasi per fitur, jika ada masalah atau kesalahan, saya bisa mengetahuinya lebih awal. Ini jauh lebih baik daripada menyadari ada kesalahan besar setelah semua proses *coding* selesai.



Tahapan Implementasi Proyek Anda

1. **Tahap 1: Perancangan Awal & Prototipe (Sudah Selesai)**
 - **Aktivitas:** Mendefinisikan ide, membuat SRS, merancang diagram (Use Case, ERD, dll.), dan membuat desain UI/UX.
 - **Output:** Dokumen perancangan yang jelas dan prototipe visual yang kita jadikan acuan.
2. **Tahap 2: Iterasi #1 - Implementasi Fitur Inti (Autentikasi & Jadwal)**
 - **Perencanaan:** Siapkan proyek Firebase, siapkan halaman login/registrasi.
 - **Pengembangan:** Mulai *coding* untuk fitur registrasi, login, dan semua fungsi pada modul **Jadwal** (tambah, lihat, edit, hapus).
 - **Pengujian:** Uji coba semua fungsi yang sudah dibuat di tahap ini untuk memastikan tidak ada *bug*.
 - **Hasil:** Di akhir tahap ini, Anda sudah memiliki aplikasi yang bisa digunakan untuk login dan mengelola jadwal.
3. **Tahap 3: Iterasi #2 - Implementasi Fitur To-Do List**
 - **Pengembangan:** Fokus *coding* hanya pada fitur **To-Do List** (tambah, lihat, filter, tandai selesai).
 - **Pengujian:** Uji coba khusus fitur To-Do List dan interaksinya dengan fitur yang sudah ada.
 - **Hasil:** Aplikasi Anda sekarang memiliki dua fitur utama yang berfungsi penuh.
4. **Tahap 4: Iterasi #3 - Implementasi Fitur Keuangan**
 - **Pengembangan:** Lakukan hal yang sama untuk fitur **Keuangan**.
 - **Pengujian:** Uji coba menyeluruh fitur Keuangan.
 - **Hasil:** Semua fitur dasar aplikasi Anda telah selesai dibuat.
5. **Tahap 5: Rilis Awal & Umpan Balik**
 - **Rilis:** Anda bisa merilis versi pertama aplikasi Anda (misalnya kepada teman-teman untuk dicoba).
 - **Umpan Balik:** Kumpulkan masukan, kritik, dan saran dari pengguna awal.
 - **Perencanaan Berikutnya:** Masukan tersebut menjadi dasar untuk perencanaan iterasi selanjutnya (misalnya, perbaikan *bug* atau penambahan fitur baru).

4. *Lain-Lain*

4.1 Github

https://github.com/Kevinardi17/Mobile_Programming.git