

## **BAB II**

### **KAJIAN PUSTAKA**

#### **A. Kajian Teori**

##### **1. Rancang Bangun**

Menurut Pressman, pembangunan atau rancang sistem adalah proses menciptakan sistem baru atau melakukan perubahan serta perbaikan pada sistem yang telah ada secara menyeluruh. Pengertian bangun atau Pembangunan sistem merujuk pada proses menciptakan sistem baru atau melakukan perubahan dan perbaikan pada sistem yang sudah ada, baik secara menyeluruh maupun sebagian (Akbar Johan dkk., 2022).

Dengan demikian, rancang bangun dapat diartikan sebagai tahapan dimana ide-ide direpresentasikan, direncanakan, dan disusun sedemikian rupa sehingga elemen-elemen terpisah dapat digabungkan menjadi suatu kesatuan yang berfungsi. Dalam konteks ini, rancang bangun melibatkan terjemahan hasil analisis ke dalam bentuk paket perangkat lunak, baik itu dengan menciptakan sistem baru atau memperbaiki sistem yang sudah ada.

##### **2. Aplikasi**

Definisi aplikasi adalah penerapan dari rancangan sistem yang digunakan untuk memproses data dengan mengikuti aturan atau ketentuan bahasa pemrograman khusus (Permatasari & Prehanto, 2013). Secara terminologi, aplikasi mengacu pada perangkat lunak yang telah dipreparasi untuk penggunaan. Perangkat lunak ini didesain khusus untuk melaksanakan fungsi tertentu bagi pengguna layanan aplikasi, dan juga dapat dimanfaatkan oleh sekelompok orang yang merupakan target pengguna yang dituju (Fansuri, 2018).

Hasan Abdurahman dan Asep Ririh Riswaya (2014) mendefinisikan aplikasi sebagai suatu program yang telah dipersiapkan untuk menjalankan perintah-perintah dari penggunanya (Wahyuni dkk.,

2022). Tujuan dari penggunaan aplikasi ini adalah untuk mencapai hasil yang akurat sesuai dengan niat pembuat aplikasi tersebut. Secara keseluruhan, aplikasi bisa diartikan sebagai solusi untuk masalah yang melibatkan pemrosesan data. Penggunaan aplikasi ini dirancang khusus sesuai dengan kebutuhan komputasi yang diinginkan dan diharapkan. Oleh karena itu, aplikasi merupakan sebuah perangkat komputer yang siap pakai, disesuaikan dengan kebutuhan pengguna dan dapat digunakan sesuai dengan kapabilitasnya.

### **3. Manajemen**

Pengertian manajemen secara umum menggambarkan rangkaian tindakan yang diambil oleh individu atau kelompok untuk mengorganisir kegiatan mereka. Tujuan utama dari manajemen ini adalah mencapai sasaran yang telah ditetapkan oleh individu atau kelompok tersebut melalui kerjasama, dengan mengoptimalkan pemanfaatan semua sumber daya yang tersedia (Tifani, 2022).

Menurut Lawrence A. Appley, sebagaimana dikutip oleh Ibnu Syamsi, manajemen adalah keterampilan menggerakkan orang untuk menyelesaikan tugas dengan tujuan tertentu. Pernyataan Soekarno mengonfirmasi bahwa manajemen adalah proses mencapai tujuan khusus melalui kerjasama dengan orang lain (Edeng, Suryana, 2017).

Dapat disimpulkan bahwa manajemen merupakan upaya untuk mengatur orang-orang agar dapat bekerja sama dalam mencapai tujuan yang telah ditetapkan, dengan mengoptimalkan keterampilan dan kerjasama mereka. Dalam konteks ini mencakup pengelolaan tugas, file, dan kolaborasi di antara pengguna aplikasi, menciptakan lingkungan belajar yang terstruktur dan efisien.

### **4. Kolaborasi**

Menurut Kamus Heritage Amerika (2000), kolaborasi adalah esensi dari bekerja sama, terutama dalam upaya menggabungkan berbagai pemikiran. Sementara itu, Jonathan (2004) mendefinisikan

kolaborasi sebagai proses interaksi berkelanjutan antara beberapa individu. (Sosiologi, 2023)

Menurut pandangan Schrage yang dipaparkan oleh Aggranoff dan McGuire (2003:4), kolaborasi adalah relasi yang didesain secara sengaja untuk memecahkan masalah dengan cara menciptakan solusi, terutama ketika dihadapkan pada keterbatasan, seperti keterbatasan informasi, waktu, dan ruang (Arrozaaq, 2016).

Para ahli memiliki berbagai pandangan tentang kolaborasi, dengan prinsip-prinsip seperti kebersamaan, kerjasama, pembagian tugas, kesetaraan, tanggung jawab, dan pertanggungjawaban. Secara umum, kolaborasi adalah hubungan antar organisasi yang mencapai tujuan bersama. Ini melibatkan pertukaran informasi, sumber daya, dan tanggung jawab dalam pengambilan keputusan, dengan tujuan menyelesaikan masalah melalui dukungan saling bantu-membantu (Dorisman dkk., 2021).

## **5. Kelas**

Menurut definisi yang diambil dari Kamus Besar Bahasa Indonesia (1995) yang dikutip oleh Sudarwan Danim, "kelas" dapat diartikan sebagai ruang di sekolah tempat belajar. Sementara itu, menurut Hornby dalam Oxford Advanced Learner's Dictionary (1986), "kelas" adalah sekelompok siswa yang diajar bersama, atau tempat di mana sekelompok siswa tersebut bertemu untuk proses pembelajaran pada waktu yang sama. Dengan kata lain, kelas merupakan lingkungan di sekolah di mana sekelompok siswa belajar bersama atau tempat di mana mereka diajar pada waktu yang bersamaan (Edeng, Suryana, 2017).

## **6. Flutter**

Flutter adalah kerangka kerja sumber terbuka dari Google untuk membangun aplikasi *multi-platform* dengan antarmuka pengguna yang baik. Dikompilasi secara asli dari satu basis kode, aplikasi Flutter dapat berjalan dengan baik di berbagai *platform* (Yudhistira, 2021). Flutter dipercaya oleh merek-merek terkenal global. Dikelola oleh komunitas

pengembang global, Flutter memungkinkan kolaborasi dalam proyek sumber terbuka, kontribusi pada ekosistem paket di pub.dev, dan menyediakan bantuan saat dibutuhkan. Menggunakan bahasa pemrograman Dart yang dioptimalkan, Flutter memungkinkan pengembangan aplikasi yang responsif dan menarik dengan kecepatan tinggi (Flutter, 2024a).

## 7. Dart

Dart adalah bahasa pemrograman serbaguna yang tidak hanya terbatas pada penggunaan dalam Flutter, melainkan juga dapat diterapkan dalam berbagai konteks lainnya. Bahasa pemrograman ini didesain untuk pengembangan aplikasi web, *mobile*, *server*, dan desktop. Dengan demikian, dapat digunakan untuk membangun berbagai jenis aplikasi sesuai kebutuhan pengembang (Dart, 2024).

Salah satu keunggulan utama Dart adalah kemampuannya untuk diterjemahkan ke dalam JavaScript. Ini memungkinkan pengembangan *front-end* web dengan pendekatan serupa seperti bahasa pemrograman lainnya. Dengan fleksibilitas ini, Dart menjadi pilihan yang sangat baik bagi pengembang yang mencari bahasa pemrograman serbaguna untuk berbagai jenis proyek pengembangan perangkat lunak (Kumar, 2023).

## 8. Firebase

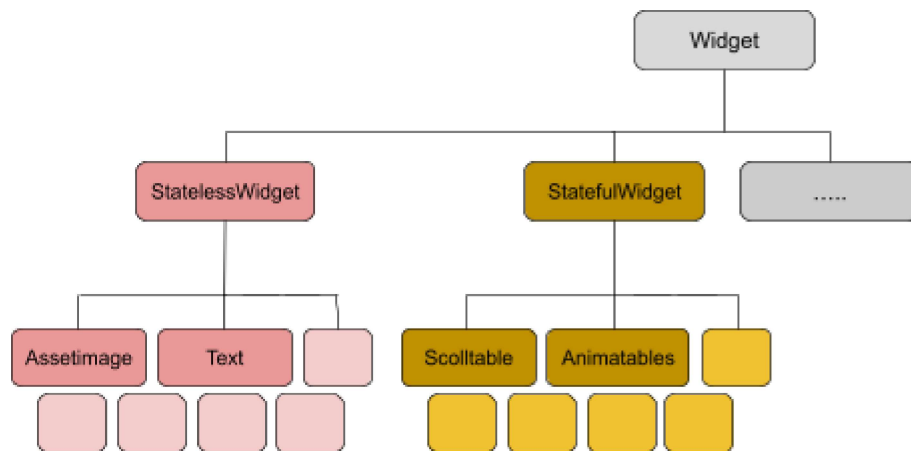
Firebase adalah layanan *backend* siap pakai dari Google, memungkinkan pengembang fokus pada pengembangan aplikasi tanpa khawatir tentang kompleksitas *backend*. Dengan fitur *real-time* database, otentikasi pengguna, dan penyimpanan file, Firebase mempercepat pembuatan aplikasi yang interaktif dan responsif secara efisien (Firebase, 2023).

Firebase menawarkan dua fitur menarik: Firebase *Remote Config* dan Firebase *Realtime Database*. Selain itu, terdapat juga Firebase *Notification*, suatu fitur pendukung yang memungkinkan aplikasi memberikan pemberitahuan kepada pengguna secara efektif (Darnita & Muntahanah, 2021).

Selain menggunakan database, Firebase juga menyediakan beragam layanan tambahan yang dimanfaatkan dalam pengembangan aplikasi ini. Beberapa layanan tersebut antara lain adalah Autentikasi dan Penyimpanan. Layanan Firebase *Storage*, misalnya, mirip dengan konsep penyimpanan awan atau *cloud storage*. Dengan layanan ini, pengembang dapat dengan mudah mengunggah dan mengunduh berkas yang diperlukan dalam aplikasi. Hal ini memungkinkan pengguna aplikasi untuk menyimpan dan mengakses data secara efisien, meningkatkan kinerja dan fungsionalitas aplikasi secara keseluruhan (Kasman & Santoso, 2022).

## 9. Widget

Widget merupakan elemen-elemen pendukung dalam Flutter, seperti *Button*, *Text*, *Icon*, dan elemen lainnya. Istilah ini merujuk pada semua komponen yang ada dalam tampilan aplikasi (Miqdad, 2023). Dalam konteks pengembangan Flutter, terdapat dua jenis widget utama: *stateless widget* dan *stateful widget*. *Stateless widget* tidak memiliki status atau data yang berubah, sehingga informasi yang ditampilkan tetap konstan sepanjang waktu. Di sisi lain, *stateful widget* memiliki kemampuan untuk mengubah status atau data selama aplikasi berjalan, memungkinkan perubahan dinamis dalam tampilan atau perilaku widget (Flutter, 2024). Kedua jenis widget ini berada pada tingkat hierarki yang sama dalam kerangka kerja Flutter seperti terlihat pada Gambar 1.1.



Gambar 1.1. Hierarki *Flutter Class Widget* (Fayzullaev, 2018)

## 10. *Material Design*

Desain *Material* adalah kerangka desain yang bersifat *open-source* yang dikembangkan oleh Google dengan kontribusi dari desainer dan pengembang. *Material 3*, versi terbaru dari desain ini, menyajikan pengalaman pengguna yang personal, adaptif, dan ekspresif. Beberapa fitur baru meliputi penggunaan warna dinamis, peningkatan aksesibilitas, serta landasan untuk tata letak layar besar dan token desain. Saat ini, Flutter sedang mengalami proses migrasi ke *Material 3* untuk meningkatkan pengalaman pengguna dalam aplikasi (Flutter, 2024).

## 11. Database

Basis data, atau database merupakan sekumpulan informasi yang tersimpan di dalam komputer secara terstruktur. Data ini dapat diperiksa dan diambil melalui program komputer. Sistem ini memungkinkan penyimpanan data yang terorganisir dengan baik, memungkinkan akses dan penggunaan data sesuai kebutuhan melalui perangkat lunak komputer (Agung dkk., 2021).

Tujuan utama dari sistem basis data adalah memungkinkan pengguna untuk membentuk pandangan abstrak terhadap data. Tujuan ini adalah untuk menyederhanakan interaksi antara pengguna dengan sistem, memungkinkan basis data menampilkan pandangan yang berbeda kepada pengguna, programmer, dan administrator (Andaru, Andri, 2018).

## 12. Android Studio

Android Studio adalah sebuah *Integrated Development Environment (IDE)* yang digunakan untuk mengembangkan aplikasi Android (Maiyana, 2018). Bahasa pemrograman utama yang digunakan adalah Java, sedangkan untuk membuat tampilan atau *layout*, digunakan bahasa XML. Android studio juga terintegrasi dengan *Android Software Development Kit (SDK)* untuk *deploy* ke perangkat android (Fikri Paturahman dkk., 2021). Android Studio menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat aplikasi Android (Developers, 2024).

### 13. SDK

Software Development Kit (SDK) merupakan kumpulan alat khusus *platform* yang diperuntukkan bagi pengembang. Dalam SDK, terdapat komponen-komponen vital seperti debugger, compiler, dan pustaka yang diperlukan untuk menulis kode yang dapat dijalankan di *platform*, sistem operasi, atau bahasa pemrograman tertentu. Seluruh kebutuhan untuk mengembangkan dan menjalankan perangkat lunak tersedia dalam satu paket. Selain itu, SDK juga dilengkapi dengan sumber daya seperti dokumentasi, tutorial, panduan, serta API dan kerangka kerja untuk mempercepat proses pengembangan aplikasi (Amazon, 2023).

### 14. Unified Modelling Language (UML)

#### a. Pengertian Unified Modelling Language (UML)

UML, atau Unified Modeling Language, adalah bahasa berbasis grafik/gambar yang digunakan untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan sistem pengembangan perangkat lunak berbasis *Object-Oriented* (OO). UML menyediakan seperangkat notasi standar yang mencakup diagram seperti diagram *use case*, diagram kelas, diagram aktifitas, dan lainnya, yang memungkinkan pengembang untuk secara jelas mendeskripsikan struktur, perilaku, dan interaksi dalam sistem perangkat lunak berbasis objek (Zufria, 2013).

UML juga dapat digunakan sebagai alat untuk mentransfer pengetahuan tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lain. Ini tidak hanya terbatas pada *developer* saja, tetapi juga dapat digunakan oleh orang bisnis dan siapa saja dapat memahami sistem dengan menggunakan UML (Intern, 2021).

#### b. Diagram-diagram dan Notasi UML


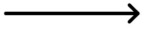
Notasi adalah metode menyampaikan informasi tentang sistem atau aplikasi dengan menggabungkan teks dan gambar. Formalisasi notasi ini mempermudah komunikasi dan dokumentasi

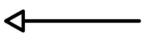




(Febriani, 2020). UML (Unified Modeling Language) merupakan salah satu notasi yang umum digunakan. Beberapa jenis diagram UML yang digunakan dalam notasi ini meliputi:

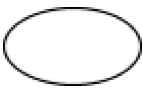
1) *Use Case Diagram*

*Use case diagram* menggambarkan relasi antara aktor dan sistem, memperlihatkan interaksi antara mereka. Diagram ini merinci interaksi antara satu atau lebih aktor dengan sistem yang sedang dibangun, menjelaskan fungsi sistem, dan menggambarkan komunikasi antara aktor dan sistem. *Use case diagram* berguna untuk menyederhanakan presentasi interaksi ini, memudahkan pemahaman oleh konsumen (Binus, 2019). Simbol-simbol yang digunakan dalam *Use Case Diagram* dijelaskan pada tabel berikut:

Tabel 1.1 Simbol – simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).

3.		<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.


8.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
----	---	-----------------	---





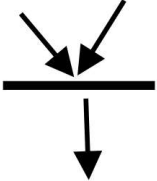
Sumber: (Mirawati, 2019)

## 2) *Diagram Activity*

Diagram aktivitas bersifat dinamis, merupakan varian dari diagram keadaan yang menggambarkan aliran aktivitas dalam suatu sistem. Berbeda dengan diagram keadaan biasa, diagram aktivitas menekankan pada urutan tindakan dari satu aktivitas ke aktivitas berikutnya. Sebagian besar dari elemen-elemen dalam diagram aktivitas adalah tindakan, dan transisi antara aktivitas-aktivitas tersebut umumnya dipicu oleh penyelesaian aktivitas sebelumnya, yang menunjukkan proses internal. Oleh karena itu, meskipun diagram aktivitas tidak secara persis menggambarkan perilaku internal sistem atau interaksi antara subsistem secara detail, namun lebih menyoroti proses umum dan alur aktivitas pada tingkat yang lebih abstrak (Zufria, 2013).

Tabel 1.2 *Diagram Activity*

No	Gambar	Nama	Keterangan
1.		<i>Actions</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama

			lain
2.		<i>Decision</i>	State dari untuk menyatakan kondisi dari suatu proses.
3.		<i>Start</i>	Bagaimana objek dibentuk atau diawali.
4.		<i>Stop</i>	Bagaimana objek dibentuk dan dihancurkan
5.		<i>Fork</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
6.		<i>Join</i>	Untuk menunjukkan kegiatan yang dilakukan/ digabungkan

Sumber: (Mirawati, 2019)

## 15. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) digunakan untuk menggambarkan entitas dan relasi dalam bentuk diagram data, memungkinkan representasi notasi menjadi format yang menjelaskan proses transaksi (Indarta dkk., 2021). Selain digunakan dalam perancangan database, ERD juga bermanfaat untuk mengatasi masalah pada database. *Debugging* database bisa menjadi tugas yang rumit, terutama jika database memiliki banyak tabel dan menggunakan *query* SQL kompleks. Dengan menggunakan ERD, skema database dapat divisualisasikan dengan jelas, memudahkan pengguna untuk mengidentifikasi dan memecahkan masalah yang muncul, menjadikan

proses debug lebih efisien dan mudah dipahami (Setiawan, 2021). ERD memiliki tiga konsep utama yaitu :

a. Entitas

Sebuah entitas dapat berupa orang, tempat, objek, atau kejadian yang dapat dianggap penting bagi sebuah organisasi atau perusahaan. Setiap entitas memiliki beberapa atribut yang mendeskripsikan karakteristik dari objek. Atribut yang ada dalam entitas harus disimpan dan dicatat dalam basis data (Pratama & Junianto, 2016). Entitas pada komponen ERD dapat dibedakan menjadi dua macam yaitu strong entity dan weak entity. Strong entity merupakan entitas yang tidak bergantung pada entitas lain atau entitas yang dapat berdiri sendiri. Sedangkan untuk weak entity merupakan entitas yang keberadaannya tergantung pada entitas lain (Togatorop dkk., 2021).

b. Atribut

Setiap entitas memiliki karakteristik tertentu yang disebut dengan atribut. Atribut berfungsi untuk mendeskripsikan karakteristik yang ada pada entitas yang disimpan dalam basis data. Berdasarkan karakteristik sifatnya, atribut dapat dibedakan menjadi beberapa jenis yaitu *simple attribute* dan *composite attribute*, *single valued attribute* dan *multi value attribute*, *derived attribute*, *key attribute*. *Primary key* adalah nama untuk atribut yang digunakan dalam mengenali suatu entitas. Atribut dalam entitas yang merupakan *primary key* adalah kode identifikasi yang bersifat unik ditunjukkan berdasarkan masing-masing *record* pada sistem. *Primary key* bertujuan untuk memberitahu lokasi untuk tiap catatan pada suatu file tentang catatan-catatan yang sama (Munif, Abdul, 2013).

c. Relasi

Relasi adalah sebuah hubungan antara dua atau lebih entitas yang saling berkaitan. Relasi pada ERD dapat digambarkan dengan menggunakan simbol belah ketupat (*diamond*). Relasi

memiliki beberapa jenis relasi yaitu *unary*, *binary*, *ternary* (Togatorop dkk., 2021).

## 16. *Wireframe*

*Wireframe* adalah elemen penting dalam pengembangan produk, membantu desainer, pengembang, dan manajer produk. Sebagai kerangka gambar, *wireframe* menghubungkan konsep struktural dengan visual desain website atau aplikasi (Fahrudin & Ilyasa, 2021). Dalam proses pengembangan, *wireframe* menyusun elemen-elemen seperti teks, gambar, dan tata letak, dikerjakan oleh UI/UX *Designer* menggunakan kertas coretan atau perangkat lunak khusus. Keberadaan *wireframe* memberikan panduan yang jelas bagi pengembang, mencegah kesulitan dan revisi berulang yang dapat menghambat proyek. Tanpanya, proyek dapat tertunda dan mengancam tenggat waktu pengembangan (Dicoding, 2024).

## B. Penelitian Relevan

Ada beberapa hasil penelitian yang relevan atau berhubungan dengan penelitian yang dilakukan oleh peneliti, diantaranya :

1. Penelitian yang dilakukan oleh Yulia Darnita dan Muntahanah dari Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Bengkulu, dengan judul “Rancang Bangun Aplikasi *Mobile* Penjadwal Perkuliahan Dengan Firebase Dengan *Realtime Notification*”. penelitian ini dilakukan untuk mengembangkan aplikasi *mobile* penjadwal perkuliahan dengan firebase dengan *realtime notification*. Aplikasi ini bertujuan untuk memberikan manfaat dalam mengingatkan jadwal atau agenda kegiatan mahasiswa, terutama yang sering lupa atau ketinggalan jadwal perkuliahan dan ujian. Aplikasi ini juga memanfaatkan kecanggihan *smartphone* android yang banyak dimiliki oleh mahasiswa di era modern ini. Peneliti menggunakan Metode Rapid Application Development (RAD). Metode ini digunakan untuk mengembangkan aplikasi *mobile* dengan cepat dan efisien, dengan tahapan-tahapan

sebagai berikut: fase perencanaan syarat-syarat, fase perencanaan, fase konstruksi, dan fase pelaksanaan. Hasil dari penelitian ini adalah aplikasi yang dapat membantu mahasiswa dalam mengingatkan jadwal atau agenda kegiatan mereka. Aplikasi ini menghadirkan pemberitahuan secara otomatis sehingga dapat mempermudah mahasiswa untuk mendapatkan jadwal perkuliahan langsung dari dosen yang bersangkutan. Namun, aplikasi ini memiliki kekurangan yaitu harus melewati proses persetujuan dan pembatasan konten yang berlaku pada aplikasi tersebut dan aplikasi hanya berjalan di phone yang ditargetkan (Darnita & Muntahanah, 2021).

2. Penelitian yang dilakukan oleh M. Yudha Putra dan Dwi Ely Kurniawan dari jurusan Teknik Informatika Politeknik Negeri Batam dengan judul “Implementasi Sistem *reminder* Jadwal pada *eLearning* Moodle Berbasis API Menggunakan *Framework* Flutter”. Penelitian ini bertujuan untuk mengembangkan aplikasi seluler berbasis Flutter yang memberikan pengguna informasi tentang jadwal kuliah dan tugas. Aplikasi ini mencakup fitur pengingat yang mengirim notifikasi tentang kuliah dan tugas melalui WhatsApp dan notifikasi dalam aplikasi. Penelitian ini menggunakan metode model waterfall, sementara pengujian menggunakan black-box *testing* dan Firebase Test Lab. Hasil penelitian ini adalah dari hasil pengujian yang menunjukkan pengembangan aplikasi ini telah berhasil. Selain itu, aplikasi ini memudahkan akses yang efisien dan mudah ke jadwal kuliah dan tugas. Hasil uji performa aplikasi menunjukkan dapat berjalan pada berbagai perangkat dan sistem operasi Android. Terutama pada perangkat *mobile* Redmi 6A menunjukkan implementasi running time dibawah 2 detik dengan penggunaan sumber daya CPU rata-rata mencapai 20%, grafis 55-60 fps dan penggunaan RAM mencapai 161 MB, sehingga disimpulkan implementasi aplikasi reminder tidak terlalu memakan banyak resource dengan spesifikasi minimal (Putra & Kurniawan, 2023).
3. Penelitian yang dilakukan oleh Rifqi Fahrudi dan Reza Ilyasa dari Program Studi Sistem Informasi Universitas Catur Insan Cendekia

dengan judul “Perancangan Aplikasi Nugas Menggunakan Metode Design Thinking dan Agile Development”. Penelitian ini bertujuan untuk menghasilkan aplikasi yang membantu mahasiswa dalam mengelola dan mengerjakan tugas. Metode Design Thinking digunakan untuk menciptakan UI/UX *Design*, sementara Agile Development digunakan untuk mengembangkan aplikasi secara keseluruhan. Penelitian ini menggunakan metode Design Thinking dan Agile Development. Design Thinking dan Agile Development adalah dua pendekatan yang saling melengkapi dalam pengembangan aplikasi, dengan fokus pada penciptaan solusi berbasis pengguna dan adaptasi cepat terhadap perubahan. Mereka bekerja bersama dalam pembuatan UI/UX *design* dan implementasi ide untuk memastikan pengembangan yang efisien dan antarmuka yang lebih baik. Hasil penelitian ini adalah berhasilnya aplikasi tersebut yang dapat berjalan dengan baik dan benar sesuai kebutuhan dan memberikan manfaat bagi para mahasiswa. Berdasarkan hasil pengujian *Test Case* memperoleh persentase 100% sesuai hasil yang diharapkan (Fahrudin & Ilyasa, 2021).

4. Penelitian yang dilakukan oleh Leni Andriani dan Mufidatul Islamiyah dari Institut Teknologi & Bisnis ASIA Malang, dengan judul “Rancang Bangun Sistem Personal *Owncloud Storage* untuk Arsip *File* Tugas Akhir Prodi Sistem Komputer Berbasis Raspberry Pi 3”. Penelitian ini menggunakan metode waterfall yang melibatkan langkah-langkah seperti pengumpulan data mengenai kebutuhan dan masalah prodi sistem komputer, perancangan sistem *ownCloud* storage, implementasi menggunakan perangkat keras seperti Raspberry Pi 3, integrasi dengan jaringan lokal dan internet melalui DDNS, serta uji coba fungsi *ownCloud* dan kinerja *server*. Proses ini diikuti dengan tahap perbaikan jika ditemukan kesalahan atau kekurangan dalam sistem. Hasil penelitian menunjukkan pembangunan sistem *ownCloud storage* sukses menggunakan Raspberry Pi 3, harddisk eksternal, router, dan aplikasi *ownCloud*. Sistem ini memungkinkan penyimpanan dan akses file tugas akhir secara online melalui jaringan lokal dan internet. Fungsi

ownCloud, termasuk *upload*, *download*, *share*, dan *delete file*, berjalan lancar. Admin dapat membuat akun *user* dan melakukan sinkronisasi data. Kecepatan maksimal saat *upload* berkas menggunakan DropBox sebesar 10.7 Mbps, Google Drive sebesar 10.1 Mbps, dan Owncloud sebesar 10.6 Mbps dari kedua perbandingan tersebut kecepatan dari owncloud hasilnya lebih tinggi dari Google Drive. Namun, kecepatan *upload* dan *download* tergantung pada *bandwidth* internet pengguna, dan kemampuan *server* terbatas oleh jumlah pengguna yang bersamaan, mempengaruhi durasi dan kinerja *server* (Andriani & Islamiyah, 2023).

5. Penelitian yang dilakukan oleh Sisi Igasari dari Prodi Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Padang dan Muhammad Adri dari Jurusan Teknik Elektronika Universitas Negeri Padang dengan judul “Rancang Bangun Aplikasi *Repository* Bahan Ajar Prodi PTI UNP Berbasis *Mobile Apps*”. Penelitian ini bertujuan untuk mengembangkan sistem dan aplikasi Android untuk mengelola Repositori Materi Pengajaran di Program Studi Pendidikan Teknik Informatika, Universitas Negeri Padang, dengan konsep *client-server*, sistem ini memberikan solusi pembelajaran seluler dengan akses materi pengajaran *online* dan *offline* (PDF dan video). Penelitian ini menggunakan metode Waterfall yang membagi program menjadi tiga bagian: model, tampilan, dan pengendali. Metode ini digunakan untuk merancang sistem, terdiri dari enam tahap: analisis sistem, desain sistem, implementasi, pengujian, implementasi, dan pemeliharaan. Penelitian ini menghasilkan pengembangan aplikasi seluler dan sistem informasi untuk manajemen repositori materi pengajaran di Program Studi Pendidikan Teknik Informatika, Universitas Negeri Padang. Aplikasi ini memungkinkan mahasiswa mengakses dan mengunduh materi pengajaran dalam format PDF dan video sesuai semester dan mata kuliah. Sistem ini menggunakan arsitektur *client-server* dengan aplikasi seluler sebagai klien dan layanan web sebagai *server* (berbasis PHP dan MySQL). Walaupun demikian, hasil ini perlu diverifikasi melalui tinjauan sejawat

dan replikasi untuk memastikan keandalan dan validitasnya (Igasari & Adri, 2021).

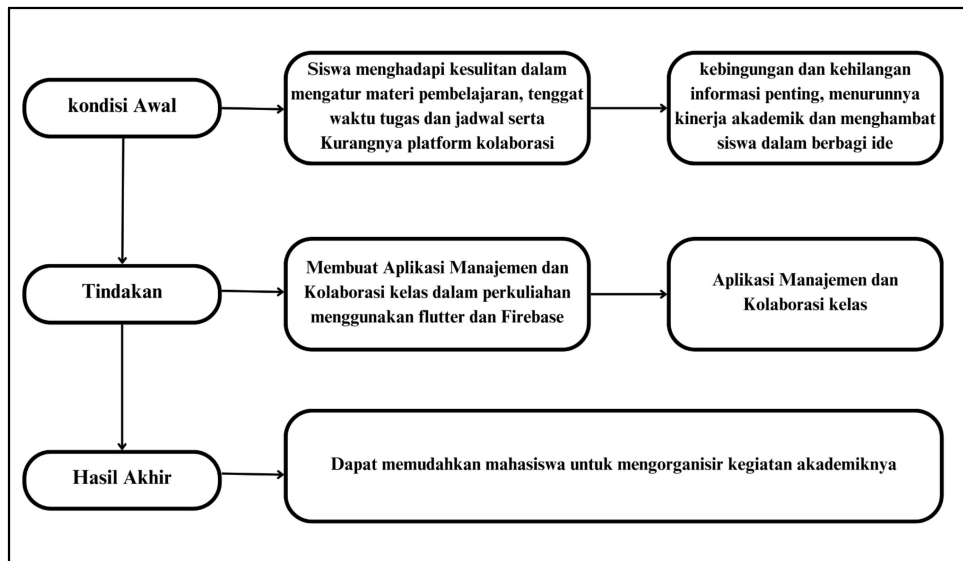
6. Penelitian yang dilakukan oleh HOD. Dr R Juliana, Naveen Kumar VG, Richard G, dan Shivadarshini P dari Department Of Information Technology Icam College Of Engineering And Technology dengan judul “Evecurate – A Smart Event Management App Using Flutter and Firebase”. Tujuan dari penelitian ini adalah untuk mempresentasikan Evecurate, sebuah aplikasi manajemen acara pintar yang menggunakan Flutter dan Firebase untuk membuat, berbagi, dan mengelola acara untuk perguruan tinggi dan universitas. Penelitian ini juga mengusulkan arsitektur Flutter baru untuk manajemen state berdasarkan konsep *Clean Architecture*. Penelitian ini menggunakan metode pengembangan aplikasi mobile *hybrid* untuk membuat Evecurate, sebuah aplikasi manajemen acara pintar menggunakan Flutter dan Firebase. Hasil dari penelitian ini adalah Arsitektur sistem untuk Evecurate dibagi menjadi 3 lapisan, yaitu: *Presentation layer*, *Business / Logic layer* dan *Data layer*. Penelitian ini menyajikan sebuah aplikasi *mobile* yang menggunakan kode QR dan teknologi GPS untuk memverifikasi kehadiran peserta acara. Penelitian ini mengklaim bahwa aplikasi tersebut meningkatkan proses pendaftaran dan kepuasan pengguna. Penelitian ini membandingkan pendekatan asli dan hibrida untuk pengembangan aplikasi seluler. Penelitian ini membahas kelebihan dan kekurangan masing-masing pendekatan dan menyarankan beberapa kriteria untuk memilih opsi terbaik (Juliana & Vg, 2021).
7. Penelitian yang dilakukan oleh Saud Maruli Panjaitan, Hendri dan Ronald Naibaho dari Fakultas Ilmu Komputer, Teknik Informatika, Universitas Dinamika Bangsa dengan judul “Perancangan Forum Diskusi Mahasiswa Berbasis Website (Studi Kasus Universitas Dinamika Bangsa Jambi)”. Tujuan dari penelitian ini adalah untuk menyusun dan membangun aplikasi forum diskusi mata kuliah pembelajaran teknik informatika universitas pasundan menggunakan *framework* laravel. sistem ini memberikan solusi terhadap permasalahan Media informatika

yang digunakan untuk diskusi adalah Learning Management System (LMS) berbasis moodle, Facebook dan WhatsApp dalam diskusi. Dimana moodle dan whatsapp saat ini lebih mengarah pada diskusi satu arah yang dipandu oleh dosen dalam kelompok diskusi, yang membuat beberapa mahasiswa merasa malu dan membuat diskusi menjadi kurang hidup. Penelitian ini menggunakan metode waterfall yang mengikuti tahapan berurutan untuk mengatur proses pengembangan sistem secara terstruktur dari mulai mengidentifikasi kebutuhan dan persyaratan sistem, merancang struktur dan tampilan antarmuka forum diskusi, membangun sistem berdasarkan desain, uji fungsionalitas sistem dan tahapan terakhir adalah pengoperasian dan pemeliharaan. Hasil dari penelitian ini adalah diskusi yang biasanya mengarah ke diskusi satu arah menggunakan moodle dan whatsapp kini dengan adanya rancangan website tersebut dapat meningkatkan kebebasan diskusi yang bertanggung jawab antar mahasiswa (Panjaitan dkk., 2022).

Persamaan penelitian sekarang dengan penelitian sebelumnya terletak pada tujuan penelitian, yaitu memberikan solusi untuk membantu mahasiswa dalam sistem pengingat jadwal, pengingat tugas, pengelolaan materi akademik dan kolaborasi antar mahasiswa, namun ada perbedaan dalam penelitian sekarang dengan penelitian sebelumnya yang terletak pada beberapa aspek. Pertama, fokus penelitian lebih spesifik pada pengembangan aplikasi manajemen dan kolaborasi kelas dalam perkuliahan, sedangkan penelitian sebelumnya hanya berfokus pada salah satunya saja. Kedua, metode penelitian yang digunakan adalah R&D (Research and Development) dengan model ADDIE, sedangkan penelitian-penelitian sebelumnya menggunakan metode pengembangan aplikasi yang berbeda seperti RAD (Rapid Application Development), waterfall, atau kombinasi Design Thinking dan Agile Development. Ketiga, penggunaan teknologi dalam pengembangan aplikasi, teknologi yang digunakan dalam penelitian terdahulu berbeda-beda seperti Flutter, Firebase, PHP, Laravel, Raspberry Pi 3 dan Java.

### C. Kerangka Berpikir

Berdasarkan latar belakang yang telah dijelaskan, muncul permasalahan yang mendorong peneliti untuk mengembangkan aplikasi yang mampu memberikan solusi kepada mahasiswa dalam mengelola kegiatan akademiknya, seperti mengelola jadwal, mengingat tugas, mengelola materi akademik dan kolaborasi antar mahasiswa. Pengembangan ini dilakukan sebagai penyelesaian atas permasalahan yang teridentifikasi di lingkungan kampus Universitas Muhammadiyah Kuningan.



Gambar 2.1 Kerangka Berpikir