

ARSITEKTUR MVVM DAN FRAMEWORK JETPACK COMPOSE PADA PENGEMBANGAN APLIKASI ANDROID (STUDI KASUS: APLIKASI SUKACOLAB)

Cahya Diantoni, Oman Komarudin, Adhi Rizal

Informatika, Universitas Singaperbangsa Karawang

Jl. HS.Ronggo Waluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat, Indonesia

2010631170060@student.unsika.ac.id

ABSTRAK

Pada era digital saat ini, mahasiswa Fakultas Ilmu Komputer semakin menyadari pentingnya mengembangkan keterampilan praktis dan pengalaman nyata dalam proyek atau dunia kerja. Namun, mahasiswa menghadapi kesulitan dalam mencari rekan dan proyek yang sesuai, sehingga perlu adanya media yang menghubungkan user untuk melakukan kolaborasi proyek tersebut. Penelitian ini mengusulkan pengembangan aplikasi Sukacolab menggunakan pola arsitektur MVVM dan *framework* Jetpack Compose dengan metode *Agile with Scrum*. Tahapan penelitian meliputi analisis kebutuhan, pembagian *backlog* ke dalam *sprint*, implementasi, dan pengujian alpha serta beta. Hasilnya adalah aplikasi Sukacolab berbasis Android yang memudahkan mahasiswa Fasilkom Unsika dalam mencari rekan dan kolaborasi proyek IT. *Alpha testing* oleh 4 narasumber ahli menunjukkan aplikasi telah menerapkan Kotlin, MVVM, dan Jetpack Compose dengan fitur yang berjalan baik. *Beta testing* melibatkan 35 mahasiswa dengan hasil keseluruhan fitur pada Sukacolab mendapatkan persentase penilaian sebesar 92,78%, sehingga telah memenuhi kegunaan (*usability*) dengan kategori “Sangat Baik”. Penelitian ini menyimpulkan bahwa aplikasi Sukacolab berhasil memenuhi kebutuhan pengguna dan memberikan pengalaman pengguna optimal.

Kata kunci : Aplikasi Android, Arsitektur MVVM, Framework Jetpack Compose

1. PENDAHULUAN

Peningkatan minat mahasiswa dalam mengembangkan keterampilan praktis dan pengalaman nyata dalam dunia kerja, khususnya di bidang Teknologi Informasi (TI), menjadi perhatian utama dalam era pendidikan digital saat ini. Mahasiswa semakin menyadari bahwa pengalaman proyek nyata bukan hanya sekadar pelengkap kurikulum, tetapi juga merupakan sarana efektif untuk mendapatkan pemahaman mendalam tentang penerapan ilmu yang telah dipelajari di dalam kelas. Hal ini terutama terlihat pada mahasiswa Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang (Fasilkom Unsika) yang semakin termotivasi untuk bekerja secara freelance dan aktif terlibat dalam proyek-proyek TI.

Dalam konteks ini, tantangan muncul terkait dengan cara mencari rekan dan mendapatkan proyek yang sesuai dengan minat dan keahlian. Untuk mengatasi kendala ini, peneliti berencana untuk mengembangkan platform bernama Sukacolab. Sukacolab dirancang sebagai tempat bagi mahasiswa Fasilkom Unsika untuk berdiskusi tentang proyek, mencari rekan kerja, dan berkolaborasi dalam proyek TI. Melalui platform ini, diharapkan mahasiswa dapat lebih mudah menemukan proyek-proyek yang sesuai dengan minat dan kemampuan mahasiswa.

Sukacolab akan dikembangkan berbasis aplikasi Android dengan pertimbangan bahwa banyaknya pengguna smartphone berbasis Android di kalangan mahasiswa Fasilkom Unsika. Dengan mengesusung platform Android, Sukacolab dapat diakses secara

lebih mudah dan cepat, mengikuti gaya hidup digital mahasiswa yang cenderung aktif menggunakan perangkat berbasis Android. Kelebihan ini tidak hanya mempermudah akses, tetapi juga memberikan pengalaman pengguna yang konsisten dan responsif.

Salah satu aspek penting dalam pengembangan aplikasi Android adalah pemilihan pola arsitektur yang tepat. Dalam hal ini, Model-View-ViewModel (MVVM) dan Jetpack Compose menjadi fokus utama. MVVM memisahkan logika bisnis dari antarmuka pengguna [1], sementara Jetpack Compose memungkinkan pengembang untuk mendefinisikan antarmuka pengguna dengan bahasa pemrograman Kotlin secara dinamis dan menarik [2].

Penelitian sebelumnya telah menunjukkan efektivitas penerapan pola arsitektur MVVM dan Jetpack Compose dalam pengembangan aplikasi Android. Contohnya, penelitian oleh Riyadhhi, et al mengenai aplikasi pengaduan masyarakat menunjukkan bahwa MVVM dapat meningkatkan kualitas layanan publik [3]. Sementara itu, penelitian oleh Dzulqarnain dan Tukino mengenai aplikasi pembelajaran bahasa Arab menunjukkan bahwa Jetpack Compose dapat digunakan untuk merancang antarmuka pengguna yang dinamis dan menarik [4]. Begitu juga dengan penelitian oleh Putra, et al mengenai aplikasi pembelajaran bahasa Bali, MVVM dan Jetpack Compose terbukti memberikan manfaat signifikan dalam pengembangan aplikasi Android [5].

Pada penelitian sebelumnya, penerapan pola arsitektur MVVM dan Jetpack Compose telah berhasil digunakan dalam berbagai konteks pengembangan

aplikasi Android, seperti aplikasi pengaduan masyarakat, aplikasi pembelajaran bahasa Arab, dan aplikasi pembelajaran bahasa Bali. Dalam konteks penelitian ini, Sukacolab dianggap sebagai objek yang relevan untuk mengilustrasikan implementasi dari pola arsitektur MVVM dan teknologi Jetpack Compose dalam pengembangan aplikasi Android untuk mendukung kolaborasi dan pencarian proyek IT bagi mahasiswa.

2. TINJAUAN PUSTAKA

2.1. Aplikasi Mobile

Aplikasi mobile adalah perangkat lunak yang dirancang khusus untuk dijalankan pada perangkat seluler seperti smartphone dan tablet. Aplikasi ini dapat memberikan beragam fungsi dan layanan kepada pengguna, mulai dari komunikasi, hiburan, produktivitas, hingga transaksi keuangan. Dengan kemajuan teknologi yang pesat, aplikasi mobile menjadi semakin penting dalam kehidupan sehari-hari, memudahkan pengguna dalam mengakses informasi, berinteraksi dengan orang lain, dan menjalankan berbagai aktivitas dengan cepat dan efisien [6].

2.1.1. Android

Android merupakan sistem operasi berbasis Linux yang dikembangkan oleh Google khusus untuk perangkat mobile seperti smartphone dan tablet. Keunggulan utamanya terletak pada sifat terbuka (open-source) yang memungkinkan pengembang untuk mengakses dan memodifikasi kode sumbernya, serta integrasi yang kuat dengan layanan Google seperti Gmail dan Google Maps. Dukungan untuk berbagai jenis aplikasi melalui Google Play Store juga membuat Android menjadi pilihan utama bagi pengguna yang menginginkan fleksibilitas dan beragamnya fitur yang ditawarkan. [7].

2.1.2. Kotlin

Kotlin merupakan produk pengembangan dari perusahaan JetBrains. Kotlin adalah sebuah bahasa pemrograman *modern* yang berjalan di atas Java *Virtual Machine* (JVM) [8].

2.1.3. MVVM

Pola Arsitektur MVVM merupakan bagian dari *Clean Architecture*. Clean Architecture adalah suatu pendekatan desain perangkat lunak yang memisahkan antara aturan bisnis inti, logika aplikasi, dan detail teknis implementasi [9]. MVVM adalah arsitektur perangkat lunak yang memisahkan Model, View, dan ViewModel, di mana ViewModel mengelola logika tampilan dan interaksi data antara Model dan View [1].

2.1.4. Jetpack Compose

Jetpack Compose adalah *toolkit* pengembangan antarmuka pengguna (UI) yang dipelajari untuk aplikasi Android, memungkinkan pengembang untuk membuat tampilan UI dengan kode Kotlin secara deklaratif dan responsif. Ini mengubah cara

pengembang membuat dan mengelola UI Android, meningkatkan produktivitas dan fleksibilitas dalam pengembangan aplikasi [2].

2.2. Metode Agile with Scrum

Metode *Agile with Scrum* adalah pendekatan pengembangan perangkat lunak yang menekankan pada kerjasama tim, iterasi cepat, dan adaptabilitas dalam merespons perubahan kebutuhan pelanggan. Dengan menggunakan iterasi *sprint* dan peran-peran yang terdefinisi dengan jelas, metode ini memungkinkan tim untuk secara efisien mengembangkan produk yang lebih adaptif dan responsif [10].

2.2.1. Product Backlog

Product Backlog adalah daftar prioritas yang berisi seluruh pekerjaan yang perlu dilakukan untuk mengembangkan sebuah produk, disusun berdasarkan kebutuhan dan nilai bisnisnya. Daftar ini terus diperbarui dan diatur menurut prioritasnya, memungkinkan tim pengembangan untuk fokus pada pekerjaan yang paling penting dan bernilai tinggi [10].

2.2.2. Sprint Planning

Sprint Planning adalah pertemuan di awal setiap sprint dalam metodologi Scrum di mana tim mengidentifikasi dan memprioritaskan pekerjaan yang akan diselesaikan selama *sprint* tersebut, serta merencanakan bagaimana pekerjaan tersebut akan dijalankan dan diselesaikan. [10].

2.2.3. Sprint

Sprint dalam metodologi *Scrum* adalah periode waktu terbatas yang biasanya berlangsung selama 1-4 minggu, di mana tim bekerja secara intensif untuk menyelesaikan pekerjaan yang telah diprioritaskan dalam *Product Backlog*. Selama *sprint*, tim fokus pada pengembangan dan pengujian fitur-fitur yang telah direncanakan, dengan tujuan menghasilkan inkrement produk yang siap digunakan [10].

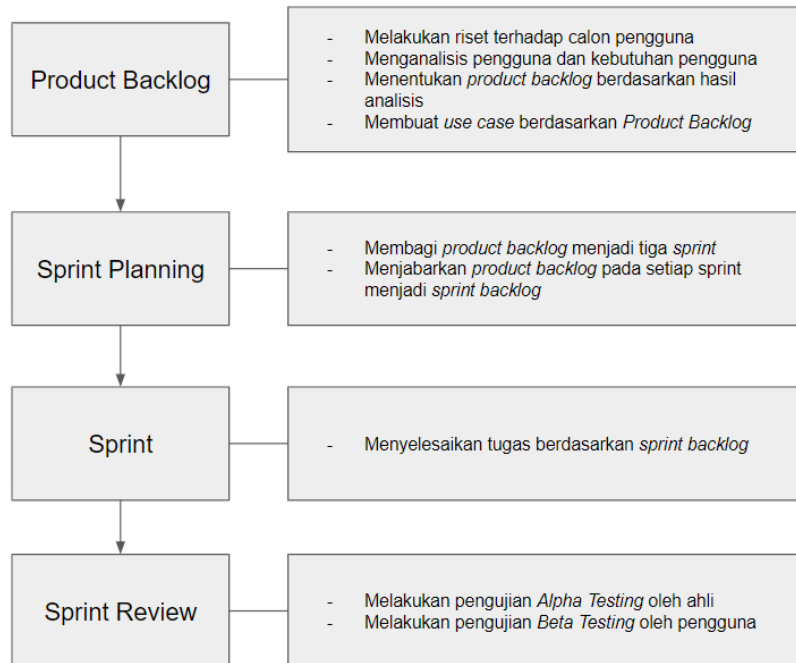
2.2.4. Sprint Review

Sprint Review adalah pertemuan di akhir setiap sprint dalam metodologi *Scrum* di mana tim memperlihatkan inkrement produk yang telah selesai kepada stakeholder, memperoleh umpan balik, dan mengevaluasi pencapaian *sprint* serta mengidentifikasi perbaikan yang diperlukan untuk *sprint* berikutnya. [10].

2.3. UML

UML (Unified Modeling Language) adalah bahasa visual yang digunakan untuk merancang dan mendokumentasikan desain sistem perangkat lunak. Ini mencakup diagram seperti diagram kelas, diagram aktivitas, dan diagram sekuen [11].

3. METODE PENELITIAN



Gambar 1. Metode penelitian

Gambar 1 merupakan metode penelitian yang digunakan pada penelitian ini. Metode penelitian yang digunakan pada penelitian ini adalah *agile with scrum*.

3.1. Product Backlog

Pada tahap ini, peneliti melakukan analisa untuk menetapkan pengguna dan kebutuhan pengguna sebagai dasar dalam pengembangan aplikasi Sukacolab. Peneliti melakukan survei yang berisi pertanyaan-pertanyaan yang telah disesuaikan dengan tujuan penelitian saat ini. Materi pada pertanyaan tersebut adalah sebagai berikut.

Tabel 1. Materi pertanyaan product backlog

No	Pertanyaan
1.	Apakah responden tertarik untuk bergabung ke suatu proyek baik itu untuk portofolio, kompetisi, ataupun pekerjaan?
2.	Apakah responden memiliki ide proyek tapi kesulitan untuk mencari rekan untuk mengerjakan proyek itu bersama?
3.	Apakah responden pernah kesulitan saat mencari rekan yang memiliki kriteria sesuai kriteriamu?
4.	Apakah kamu pernah ingin mengikuti suatu kompetisi atau lomba tetapi belum memiliki tim/rekan?
5.	Apakah kamu pernah mendapatkan suatu project dari <i>client</i> tetapi bingung untuk mengerjakannya karena belum berpengalaman atau belum memiliki rekan?

Materi pertanyaan di atas berdasarkan pada tujuan penelitian saat ini, yaitu mengetahui ketertarikan responden dan mengetahui seberapa banyak responden yang mengalami permasalahan serupa seperti yang ingin diselesaikan pada penelitian saat ini.

3.2. Sprint Planning

Pada tahap ini, peneliti melakukan pembagian terhadap *product backlog* ke dalam beberapa *sprint*. Peneliti membagi *product backlog* menjadi beberapa *sprint* dengan waktu pengerjaan setiap *sprint* berkisar antara 2-3 minggu. Hasil dari tahap ini dijadikan sebagai acuan waktu dalam pengembangan aplikasi Sukacolab.

3.3. Sprint

Pada tahap *Sprint*, peneliti melakukan tugas atau pekerjaan yang telah dijabarkan pada tahap sebelumnya. Peneliti mengimplementasikan hasil analisis kebutuhan pada tahap *Product Backlog* sesuai dengan perencanaan yang telah dilakukan pada tahap *Sprint Planning*. Implementasi tersebut berupa menjalankan tugas dalam pengembangan aplikasi Sukacolab pada masing-masing *Sprint*.

3.4. Sprint Review

Pada tahap *Sprint Review*, peneliti memperlihatkan hasil dari setiap tugas yang telah diselesaikan. Selanjutnya, dilakukan *alpha testing* terhadap setiap *sprint* yang telah selesai, serta *beta testing* terhadap keseluruhan fitur aplikasi.

Alpha testing dilakukan pada akhir setiap *sprint*. Pengujian ini dilakukan oleh narasumber ahli untuk melakukan validasi terhadap struktur internal dan kode, termasuk bahasa, arsitektur, dan framework yang digunakan, serta menguji fungsionalitas fitur aplikasi Sukacolab pada setiap *sprint*. Berdasarkan rumusan, batasan, dan tujuan penelitian ini, peneliti membuat daftar materi pertanyaan yang sesuai untuk diajukan kepada narasumber ahli saat melakukan *alpha testing*.

Setelah *alpha testing* dan pengembangan aplikasi Sukacolab selesai, dilakukan *beta testing* oleh calon pengguna secara umum untuk menguji fungsionalitas seluruh fitur aplikasi Sukacolab dari perspektif pengguna akhir. Pada tahap ini calon pengguna melakukan penilaiannya terhadap setiap fitur aplikasi Sukacolab dengan skala 1 - 5.

4. HASIL DAN PEMBAHASAN

Berikut adalah hasil dan pembahasan dari penelitian yang telah dilakukan.

4.1. Hasil

Metode Penelitian berdasarkan pada rancangan penelitian yaitu mengembangkan aplikasi Sukacolab sebagai platform kolaborasi proyek bagi mahasiswa Fasilkom Unsika dengan menggunakan Arsitektur MVVM dan *Framework* Jetpack Compose sesuai dengan metodologi penelitian yang sudah ditentukan yaitu menggunakan Agile with Scrum.

4.1.1. Product Backlog

Pada tahap *Product Backlog*, melakukan riset terhadap calon pengguna, menganalisis kebutuhan pengguna dan kebutuhan sistem, membuat list daftar fitur serta membuat use case dari setiap fitur aplikasi Sukacolab.

4.1.2. Melakukan Riset terhadap Calon Pengguna

Penelitian ini dimulai dengan survei Google Form terhadap mahasiswa Fasilkom Unsika untuk memahami masalah kolaborasi proyek IT mahasiswa. Dari hasil survei, 76% responden memiliki masalah yang relevan dengan tujuan penelitian. Selanjutnya, peneliti memilih dua responden dengan masalah paling sesuai untuk wawancara.

Dari hasil wawancara didapatkan user stories berupa fitur yang dibutuhkan adalah:

- Fitur identifikasi pemilik akun (Login, Register, Logout, Kelola Profil, Pengaturan Akun).
- Fitur mencari rekan (Publikasi Proyek, Validasi Proyek oleh admin).
- Fitur melihat, mencari, dan bergabung ke proyek (List Proyek, Pencarian Proyek, Melamar ke Proyek).

4.1.3. Menganalisis Pengguna dan Kebutuhan Pengguna

Hasil analisa kebutuhan pengguna dan sistem pada aplikasi Sukacolab adalah sebagai berikut: Terdapat 2 peran pengguna pada aplikasi Sukacolab yaitu, user dan admin. Dengan kebutuhan pengguna mencakup, fitur registrasi akun, fitur *login*, fitur pengaturan akun, fitur mengubah informasi profil user, fitur mencari rekan dengan mempublikasikan proyek, fitur melihat dan mencari proyek, fitur bergabung ke proyek yang diminati, fitur review proyek oleh admin, fitur *logout* untuk user dan admin.

4.1.4. Menentukan Product Backlog Berdasarkan Hasil Analisis

Setelah menentukan kebutuhan pengguna dan sistem, langkah selanjutnya adalah menentukan *Product Backlog*. *Product Backlog* berisi fitur-fitur aplikasi yang akan dikembangkan.

Tabel 2. Product backlog

No	Nama Fitur	Pengguna
1.	Login dan register	User dan admin
2.	Pengaturan akun	User dan admin
3.	Profil pengguna	User
4.	Memublikasikan proyek	User dan admin
5.	Mencari proyek	User dan admin
6.	Join proyek	User
7.	Review proyek	Admin
8.	Logout	User dan admin

Tabel 3.1 menjelaskan mengenai product backlog yang telah dibuat pada penelitian ini. Pada aplikasi ini terdapat 8 fitur yang dikembangkan dengan dua role pengguna yaitu *user* dan *admin*.

4.1.5. Sprint Planning

Peneliti membagi beban kerja dari Product Backlog dengan mempertimbangkan waktu, kompleksitas, prioritas, dan ketergantungan antara item. Pembagian sprint dilakukan untuk mempermudah pemantauan dan meningkatkan transparansi dalam pengembangan aplikasi Sukacolab. Hal ini memastikan setiap tugas berkontribusi pada tujuan pengembangan secara keseluruhan sesuai dengan rancangan yang sudah dibuat.

4.1.6. Membagi Product Backlog Menjadi Beberapa Sprint

Terdapat tiga Sprint pada penelitian ini untuk memaksimalkan hasil yang sesuai dengan harapan, masing-masing *Sprint* dilakukan selama 2 – 3 Minggu. Berikut adalah pembagian *product backlog* menjadi masing masing sprint.

Tabel 3. Hasil sprint planning

No	Product Backlog	Sprint	Estimasi Waktu
1	Login dan register	Sprint 1	3 Minggu
2	Logout		
3	Mengelola Profil pengguna		
4	Melakukan Pengaturan akun	Sprint 2	2 Minggu
5	Memublikasikan proyek		
6	Review proyek	Sprint 3	2 Minggu
7	Melihat dan mencari proyek		
8	Join Project		

Penetapan *sprint planning* pada tabel 3 didasarkan pada kesanggupan peneliti dalam mengorganisir dan mengeksekusi pekerjaan yang diperlukan untuk mencapai tujuan *sprint*. Ini

melibatkan pembagian *product backlog* menjadi *sprint-sprint* yang dapat diselesaikan dalam periode waktu tertentu, dengan memperhatikan estimasi kompleksitas, kapasitas peneliti, dan prioritas tugas.

4.1.7. Sprint Ketiga

Setelah membuat *product backlog* dan membaginya ke setiap *sprint*, selanjutnya adalah mengerjakan tugas yang ada di *product backlog* pada *sprint* ketiga.

4.1.8. Sprint Backlog

Sesuai dengan *sprint planning*, pada *sprint* ketiga ini peneliti mengerjakan fitur melihat dan mencari proyek serta *Join Project*.

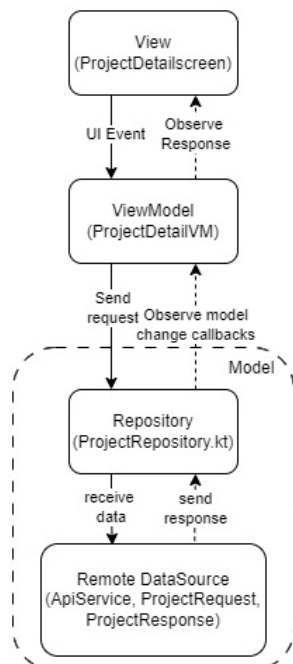
Tabel 4. Sprint Backlog Ketiga

No	Sprint Backlog
1.	Membuat desain sistem.
2.	Membuat desain database.
3.	Membuat desain antarmuka.
4.	Membuat API.
5.	Menerapkan arsitektur MVVM.
6.	Implementasi desain antarmuka menggunakan framework Jetpack Compose
7.	Integrasi API dengan logic dan antarmuka aplikasi.

Tabel 4. merupakan *sprint backlog* berisi daftar tugas yang harus dikerjakan pada *sprint* ketiga. Terdapat tujuh tugas yang harus dikerjakan selama menjalankan *sprint* ketiga pada penelitian ini.

4.1.9. Sprint

Berikut hasil pengerjaan *sprint* ketiga sesuai dengan tugas pada *sprint backlog* yang telah dibuat sebelumnya.



Gambar 2. Grafik arsitektur MVVM

Gambar 2 merupakan grafik sebagai gambaran dari penerapan arsitektur MVVM pada fitur bergabung ke proyek. Pada fitur bergabung ke proyek telah menggunakan penerapan arsitektur MVVM dengan membagi *source code* pengembangan aplikasi Sukacolab menjadi tiga bagian yaitu Model, View dan ViewModel.

4.2. Model

Model pada fitur bergabung ke proyek terdapat pada bagian *ProjectRequest*, *ProjectResponse*, *ApiService*, dan *ProjectRepository*. *ProjectRequest* digunakan sebagai model data class untuk fitur bergabung ke proyek dalam aplikasi Sukacolab. Dalam model ini, terdapat atribut-atribut seperti nama proyek, posisi yang tersedia, lokasi proyek, tipe proyek, status pembayaran, waktu atau deadline proyek, deskripsi proyek, dan persyaratan yang diperlukan untuk bergabung ke proyek tersebut.

ProjectResponse adalah model data digunakan untuk mengelola respons dari API terkait fitur bergabung ke proyek pada aplikasi Sukacolab. Model ini memiliki dua atribut utama, yaitu message yang berisi pesan respons dari server dan data yang berupa daftar proyek yang tersedia. Data proyek direpresentasikan sama dengan atribut pada *ProjectRequest* di paragraf sebelumnya.

```

@POST("/api/project/join/{projectId}")
fun joinProject(
    @Header("Authorization") token: String,
    @Path("projectId") projectId: String,
): Call<BaseResponse>
    
```

Gambar 3. Potongan code ApiService

ApiService adalah sebuah interface yang digunakan untuk menentukan *endpoint* dan parameter yang diperlukan untuk melakukan panggilan API terkait fitur bergabung ke proyek pada aplikasi Sukacolab. Fungsi utamanya adalah untuk mengonfigurasi jenis permintaan (POST dalam kasus ini), menentukan path dari endpoint API berdasarkan ID proyek yang ingin diikuti, serta menyediakan header *Authorization* yang digunakan untuk autentikasi pengguna. Dengan menggunakan interface ini, aplikasi Sukacolab dapat berkomunikasi dengan server secara efisien untuk memproses permintaan pengguna yang ingin bergabung ke proyek tertentu, dan mendapatkan respons dari server dalam bentuk *BaseResponse* yang mencakup informasi tentang keberhasilan atau kegagalan permintaan tersebut.

```

fun getDetailProject(projectId: String) = flow { this: FlowCollect
    val token = authPreferences.getAuthToken()
    Log.d(tag: "Hit API Experience", msg: "get Experience")
    emit(apiService.getDetailProject(
        token = "Bearer $token",
        projectId = projectId,
    ).data)
}.flowOn(Dispatchers.IO)
    
```

Gambar 4. Potongan code ProjectRepository

ProjectRepository adalah bagian dari model pada aplikasi Sukacolab yang bertanggung jawab untuk mengelola logika akses data terkait fitur bergabung ke proyek. Fungsi `getDetailProject` di dalamnya digunakan untuk mendapatkan detail proyek berdasarkan ID proyek yang diberikan. Saat fungsi ini dipanggil, ia akan memperoleh token autentikasi dari `AuthPreferences`, kemudian mengirimkan permintaan API menggunakan `ApiService` untuk mendapatkan data detail proyek dengan header `Authorization` yang berisi token tersebut. Hasil respons dari server berupa data proyek akan dikirimkan kembali ke tampilan melalui `flow`, memungkinkan pengguna untuk melihat informasi detail proyek sebelum memutuskan untuk bergabung ke proyek tersebut dalam aplikasi Sukacolab.

4.3. View

View pada aplikasi Sukacolab dikembangkan dengan menggunakan `framework` Jetpack Compose. Fitur bergabung ke proyek terdapat di bagian `ProjectDetailScreen`, yang bertanggung jawab untuk menampilkan detail proyek dan memberikan opsi kepada pengguna untuk bergabung ke proyek tersebut.

```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun ProjectDetailScreen(
    navController: NavController,
    projectId: String,
){
```

Gambar 5. Potongan `code` anotasi Jetpack Compose

Potongan kode tersebut menunjukkan penggunaan `framework` Jetpack Compose dalam membuat antarmuka pengguna (UI) untuk layar detail proyek pada aplikasi. Anotasi `@OptIn(ExperimentalMaterial3Api::class)` mengindikasikan penggunaan fitur eksperimental terkait `Material Design` versi 3, sementara `@Composable` sebelum fungsi `ProjectDetailScreen` menandakan bahwa layar ini dibuat menggunakan komposisi UI deklaratif dari Jetpack Compose. Dengan Jetpack Compose, pengembang dapat dengan mudah mendefinisikan tampilan dan perilaku UI berdasarkan input dan data yang diberikan, menjadikan pengembangan UI menjadi lebih modern, efisien, dan mudah dipelajari.

Potongan kode pada gambar 6, merupakan contoh penerapan `framework` Jetpack Compose untuk membuat sebuah tombol (`Button`) pada antarmuka pengguna (UI) dalam aplikasi. Dalam kode tersebut, menggunakan komponen `Button()` yang memiliki properti `onClick` untuk menentukan aksi yang dijalankan ketika tombol ditekan, `modifier` untuk mengatur lebar dan tinggi tombol, `colors` untuk menentukan warna latar belakang tombol, dan `shape` untuk memberikan bentuk sudut bulat pada tombol. Di

dalam komponen `Button()`, terdapat komponen `Text()` yang digunakan untuk menampilkan teks `"Join Project"` pada tombol dengan warna teks putih. Dengan menggunakan Jetpack Compose, pengembang dapat membuat tombol dengan mudah dan mendefinisikan berbagai properti serta gaya tampilan sesuai kebutuhan UI aplikasi.

```
Button(
    onClick = {
        viewModel.joinProject(projectId)
    },
    modifier = Modifier
        .fillMaxWidth()
        .height(50.dp),
    colors = ButtonDefaults.buttonColors(
        containerColor = MaterialTheme.colorScheme.secondary
    ),
    shape = RoundedCornerShape(8.dp)
) { this: RowScope
    Text(text = "Join Project", color = Color.White)
}
```

Gambar 6. Membuat `button` Jetpack Compose



Gambar 7. Halaman `Detail Project`

Gambar 7. merupakan hasil implementasi halaman `Detail Project` menggunakan `framework` Jetpack Compose untuk fitur bergabung ke proyek pada aplikasi Sukacolab.

4.4. ViewModel

```
fun getDetailProject(projectId: String) = viewModelScope.launch {
    projectRepo.getDetailProject(projectId)
        .onStart { this: FlowCollector<DetailProject>
            responseDetail.value = DetailProjectUIState.Loading
        }.catch { this: FlowCollector<DetailProject>
            responseDetail.value = DetailProjectUIState.Failure(it)
        }.collect { it: DetailProject
            responseDetail.value = DetailProjectUIState.Success(it)
        }
}
```

Gambar 8. Potongan `code` `ProjectDetailViewModel`

Potongan *code* ViewModel pada gambar 8. `getDetailProject`, dirancang untuk mendapatkan detail proyek yang diperlukan dalam konteks fitur bergabung ke proyek pada aplikasi. Pada saat dipanggil, fungsi ini menggunakan `viewModelScope.launch` untuk memulai proses asinkron yang berkomunikasi dengan `projectRepo` (Repository) untuk mengambil detail proyek berdasarkan ID proyek yang diberikan. Selama proses ini, berbagai keadaan UI (UI states) ditangani menggunakan `LiveData` atau `StateFlow`, seperti menampilkan indikator loading ketika proses dimulai (`DetailProjectUiState.Loading`), menangkap dan menangani kesalahan jika ada (`DetailProjectUiState.Failure`), dan mengirimkan hasil yang berhasil (`DetailProjectUiState.Success`) untuk ditampilkan pada UI. Dengan pendekatan ini, view model memisahkan logika bisnis dari tampilan dan memfasilitasi pengelolaan status UI secara bersih dan terstruktur dalam konteks fitur bergabung ke proyek pada aplikasi.

4.5. Sprint Review

Peneliti telah menyelesaikan seluruh fitur pada pada setiap sprint. *Sprint review* terdiri dari dua tahapan yaitu *alpha testing* yang dilakukan oleh narasumber ahli dan *beta testing* yang dilakukan oleh calon pengguna.

4.6. Alpha Testing

Alpha testing dilakukan oleh empat narasumber ahli yang telah dipilih berdasarkan pengalaman ahli dalam pengembangan aplikasi Android. Narasumber ahli diminta melakukan validasi terhadap kode, khususnya terkait arsitektur MVVM dan framework Jetpack Compose, serta melakukan pengujian fungsionalitas aplikasi. Berikut adalah hasil dari *sprint review* yang dilakukan oleh narasumber ahli terhadap pengembangan aplikasi dalam penelitian ini.

Tabel 5. Hasil Validasi Ahli

No	Pertanyaan	R1	R2	R3	R4
1.	Penerapan arsitektur MVVM pada pengembangan aplikasi Sukacolab telah sesuai.	☑	☑	☑	☑
2.	Penerapan <i>framework</i> Jetpack Compose pada pengembangan aplikasi Sukacolab telah sesuai.	☑	☑	☑	☑
3.	Fungsionalitas setiap fitur pada aplikasi Sukacolab berjalan dengan baik.	☑	☑	☑	☑

Kesimpulan dari *alpha testing* yang diberikan oleh narasumber ahli menegaskan bahwa pengembangan aplikasi Sukacolab telah berhasil menerapkan arsitektur MVVM dan framework Jetpack Compose dengan baik. Ini terbukti dari fungsionalitas yang baik dari setiap fitur yang dievaluasi selama alpha testing. Dengan demikian, dapat dipastikan

bahwa arsitektur MVVM dan penggunaan Jetpack Compose telah sesuai dengan standar yang diharapkan dan mendukung kualitas serta kinerja aplikasi Sukacolab dengan baik. Sehingga sprint dinyatakan berhasil dan tahap pengembangan aplikasi Sukacolab pada penelitian ini akan dilanjutkan dengan *beta testing*.

4.7. Beta Testing

Pada tahap Beta Testing, calon pengguna secara tertutup menguji fungsionalitas seluruh fitur aplikasi Sukacolab. Peneliti meminta alamat *email* mahasiswa Fasilkom Unsika, mendaftarkan calon pengguna sebagai tester melalui Google Play Console, dan mengirimkan undangan tautan. Setelah menginstall aplikasi, calon pengguna mencobanya dan menerima kuesioner review fitur aplikasi Sukacolab. Kuesioner berisi pertanyaan tentang penilaian terhadap setiap fitur aplikasi. Berikut adalah materi pertanyaan untuk beta testing terhadap fitur fitur yang telah dikembangkan oleh peneliti.

Tabel 6. Pertanyaan Beta Testing

No	Materi Pertanyaan
1	Penilaian terhadap fitur <i>Login</i> dan <i>Register</i>
2	Penilaian terhadap fitur <i>Logout</i>
3	Penilaian terhadap fitur kelola profil
4	Penilaian terhadap fitur pengaturan akun
5	Penilaian terhadap fitur memublikasikan proyek
6	Penilaian terhadap fitur melihat list proyek
7	Penilaian terhadap fitur cari proyek
8	Penilaian terhadap fitur bergabung ke proyek

Kuesioner terdiri dari delapan pertanyaan dengan skala Likert 1-5 yang digunakan untuk mengukur persepsi dan pengalaman calon pengguna terhadap aplikasi Sukacolab. Hasil dari kuesioner ini akan dihitung dan dianalisis untuk menyimpulkan hasil dari penelitian.

Tabel 7. Hasil Beta Testing

R	Pertanyaan								TTL	Maks
	1	2	3	4	5	6	7	8		
1	5	5	5	5	5	5	5	5	40	40
2	5	5	5	5	5	5	5	5	40	40
3	5	5	4	4	5	4	4	4	35	40
4	5	5	5	5	5	5	5	5	40	40
5	3	4	3	4	3	4	4	3	28	40
6	3	5	5	5	5	4	5	4	36	40
7	5	5	5	5	5	5	5	5	40	40
8	3	4	3	4	4	4	4	4	30	40
9	4	4	5	4	4	5	5	5	36	40
10	5	5	5	5	5	5	5	5	40	40
11	5	5	5	5	5	5	5	5	40	40
12	5	5	5	5	5	5	5	5	40	40
13	5	5	5	5	3	5	5	5	38	40
14	4	5	2	4	4	4	4	4	31	40
15	5	5	5	5	5	5	5	5	40	40
16	5	5	5	5	5	5	5	5	40	40
17	5	5	5	5	5	5	5	5	40	40
18	4	4	4	5	4	4	4	4	33	40
20	5	5	3	5	5	5	5	5	38	40
21	5	5	5	5	5	5	5	5	40	40

R	Pertanyaan								TTL	Maks
	1	2	3	4	5	6	7	8		
22	5	5	5	5	5	5	5	5	40	40
23	5	5	4	5	5	4	4	4	36	40
24	5	4	4	5	4	5	4	4	35	40
25	4	4	4	4	4	4	4	4	32	40
26	4	5	5	4	5	5	4	5	37	40
27	5	5	5	5	5	5	5	5	40	40
28	5	5	5	4	4	4	5	5	37	40
29	4	4	4	4	3	3	4	5	31	40
30	5	4	5	5	4	4	5	4	36	40
31	5	5	4	5	5	5	5	5	39	40
32	4	4	4	3	5	4	3	4	31	40
33	5	5	5	5	5	5	5	5	40	40
34	5	5	5	5	5	5	5	5	40	40
35	5	5	5	5	5	5	5	5	40	40
Jumlah									1299	1400

Responden terdiri dari 35 orang dengan pertanyaan sebanyak 8 pertanyaan dan skor total maksimal sebanyak 1400. Jumlah total skor yang didapatkan dari responden adalah sebanyak 1299. Skor tersebut lalu dimasukkan ke dalam rumus di atas. Berikut adalah hasil perhitungan dari skor yang sudah didapat :

$$Y = (\sum(N.R)) / (\text{Skor Ideal}) \times 100\%$$

$$Y = 1299 / 1400 \times 100\% = 92,78 \%$$

Didapatkan hasil perhitungan persentase adalah sebesar 92,78%. Berdasarkan hasil tersebut dapat disimpulkan bahwa menurut sudut pandang responden, aplikasi Sukacolab telah memenuhi kegunaan (*usability*) dengan kategori “Sangat Baik”.

4.8. Pembahasan

Penelitian menghasilkan aplikasi Sukacolab untuk mahasiswa Fasilkom Unsika dalam mencari rekan dan proyek IT. Menggunakan arsitektur MVVM dan *framework* Jetpack Compose, aplikasi ini dikembangkan dengan metode *Agile with Scrum*. Riset calon pengguna dilakukan melalui kuesioner dan wawancara untuk membuat *Product Backlog* dan *Sprint Planning*. Setelah tiga *sprint*, aplikasi diuji dengan *alpha testing* oleh ahli, yang menyimpulkan bahwa semua fitur berfungsi baik. *Beta testing* dilakukan oleh 35 mahasiswa Fasilkom Unsika, menghasilkan kelayakan aplikasi sebesar 92,78% dengan kategori "Sangat Baik".

5. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan yaitu “Arsitektur MVVM dan Framework Jetpack Compose pada Pengembangan Aplikasi Android dengan Studi Kasus Aplikasi Sukacolab” didapatkan kesimpulan sebagai berikut : Penelitian menghasilkan aplikasi Sukacolab Android untuk mempermudah mahasiswa Fasilkom Unsika mencari rekan kolaborasi proyek IT. Alpha testing melibatkan 4 penguji ahli yang menyimpulkan bahwa pengembangan aplikasi menggunakan Kotlin, MVVM, dan Jetpack Compose, serta fiturnya berjalan dengan baik untuk dilakukan beta testing. *Beta testing* melibatkan 35 mahasiswa

Fasilkom Unsika. Berdasarkan pendapat responden yang terlibat menghasilkan persentase sebesar 92,78% terhadap keseluruhan fitur aplikasi Sukacolab.

Berdasarkan kesimpulan di atas, peneliti mengajukan beberapa saran sebagai referensi untuk pengembangan penelitian dan aplikasi Sukacolab yang akan dilakukan selanjutnya : Pada pengembangan selanjutnya, peneliti menyarankan untuk membuat halaman website yang berisi pengenalan dan cara penggunaan aplikasi Sukacolab. Pada pengembangan selanjutnya, peneliti menyarankan untuk mengembangkan aplikasi Sukacolab menjadi *multi platform*. Pada pengembangan selanjutnya, peneliti menyarankan untuk menambahkan fitur-fitur menarik ke aplikasi Sukacolab agar dapat menarik banyak pengguna.

DAFTAR PUSTAKA

- [1] B. Rizki, P. Surya, A. Putra Kharisma, and N. Yudistira, “Perbandingan Kinerja Pola Perancangan MVC, MVP, dan MVVM Pada Aplikasi Berbasis Android (Studi kasus : Aplikasi Laporan Hasil Belajar Siswa SMA BSS),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 4, no. 11, pp. 4089–4095, 2020, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [2] B. Asefa, “Building Android Component Library Using Jetpack Compose,” no. March, p. 42, 2022.
- [3] I. M. Riyadhhi, Intan Purnamasari, and Kamal Prihandani, “Penerapan Pola Arsitektur Mvvm Pada Perancangan Aplikasi Pengaduan Masyarakat Berbasis Android,” *INFOTECH journal*, vol. 9, no. 1, pp. 147–158, 2023, doi: 10.31949/infotech.v9i1.5246.
- [4] F. Dzulqarnain, “Computer Based Information System Journal RANCANG BANGUN APLIKASI BELAJAR ARAB UNTUK ANDROID MENGGUNAKAN JETPACK COMPOSE DAN KOTLIN,” *Cbis Journal*, vol. 11, no. 01, pp. 25–35, 2023, [Online]. Available: <http://ejournal.upbatam.ac.id/index.php/cbishttp://ejournal.upbatam.ac.id/index.php/cbis>
- [5] I. P. R. P. Putra and H. Tolle, “Pengembangan Aplikasi Pembelajaran Bahasa Bali berbasis Android menggunakan MVVM Architecture dan Jetpack Compose,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 5, pp. 2205–2214, 2023, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [6] F. Y. Rahman, “Penerapan Metodewaterfall Pada Pengembangan Aplikasi Sistem Informasi Jasa Dan Penjualan,” *Technologia: Jurnal Ilmiah*, vol. 12, no. 2, p. 125, 2021.
- [7] Ricu Sidiq and Najuah, “Development of Android-based interactive E-modules in teaching and learning strategy courses,” *Journal of History Education*, vol. 9, no. 1, pp. 1–14, 2020, [Online]. Available:

- <https://journal.unj.ac.id/unj/index.php/jps/article/view/13650>
- [8] Arafat Febriandirza, "Perancangan Aplikasi Absensi Online Dengan Menggunakan Bahasa Pemrograman Kotlin," *Pseudocode*, vol. 6, no. 1, pp. 53–59, 2019, [Online]. Available: www.ejournal.unib.ac.id/index.php/pseudocode%0APERANCANGAN
- [9] M. B. Sinatria, Oman Komarudin, and Kamal Prihamdani, "Penerapan Clean Architecture Dalam Membangun Aplikasi Berbasis Mobile Dengan Framework Google Flutter," *INFOTECH journal*, vol. 9, no. 1, pp. 132–146, 2023, doi: 10.31949/infotech.v9i1.5237.
- [10] D. B. Kinasih, "Pengembangan Sistem Informasi Pengelolaan Kinerja Karyawan (Studi Kasus: Modena Strategy System)," *Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi*, vol. 1, no. 1, pp. 60–65, 2021, doi: 10.20885/snati.v1i1.8.
- [11] T. Arianti, A. Fa'izi, S. Adam, and Mira Wulandari, "Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram Uml (Unified Modelling Language)," *Jurnal Ilmiah Komputer ...*, vol. 1, no. 1, pp. 19–25, 2022, [Online]. Available: <https://journal.polita.ac.id/index.php/politati/article/view/110/88>