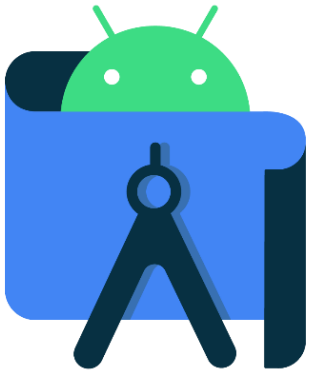
BAB 3
TINJAUAN PUSTAKA

3.1 Android Studio

Android Studio adalah sebuah Integrated Development Environment resmi untuk pengembangan aplikasi Android yang dibuat berdasarkan IntelliJ IDEA. Android Studio digunakan sebagai *code editor* dengan fungsionalitas pendukung seputar pengembangan aplikasi Android. Salah satu fitur yang disediakan oleh Android Studio yaitu *Gradle-based build system* yang memudahkan proses *build* aplikasi serta penambahan *third party library* untuk mempercepat proses pengembangan.

Android Studio menyediakan *emulator* dengan berbagai macam jenis perangkat asli yang membantu pengembang melakukan uji coba terhadap berbagai jenis perangkat dengan ukuran layar, versi OS, dan sensor virtual Android yang beragam. Penyusunan tampilan aplikasi dapat dilakukan dan dilihat langsung hasilnya melalui *side-panel* yang disediakan. Android Studio juga memiliki *debugger* komprehensif seperti *local database inspector* untuk melihat isi *database* lokal, *layout inspector* untuk melihat struktur tampilan, dan *step-by-step debugger* dengan *breakpoint* yang menunjang proses eliminasi *bug*.

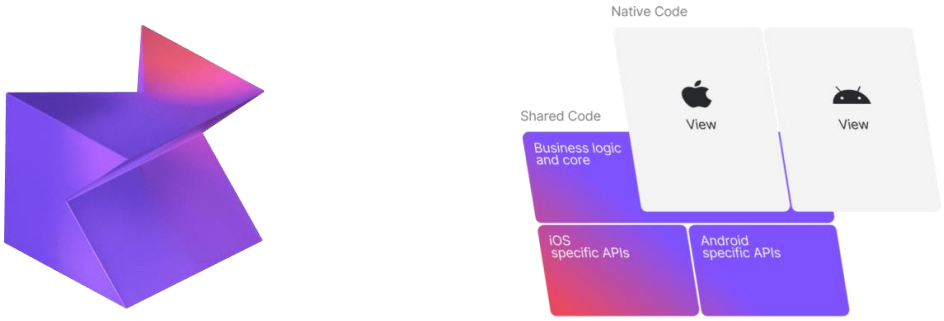


Gambar 3.1 Logo Android Studio

3.2 Kotlin Multiplatform Mobile

Kotlin Multiplatform Mobile (KMM) adalah sebuah Software Development Kit (SDK) dibuat oleh Google dan Kotlin JetBrains yang memasuki tahap Alpha pada bulan Agustus 2020. KMM memperbolehkan adanya satu *codebase* berisi *business logic* yang dapat digunakan oleh Android dan iOS. KMM memperbolehkan *codesharing* sehingga mengeliminasi *redundant code* pada penulisan *business logic* antara Android dan iOS. Pengembang hanya perlu menuliskan *platform-specific* code pada masing-masing *platform* seperti pembuatan Native UI ataupun dengan API khusus pada *platform* tersebut. Beberapa *library* esensial juga tersedia untuk menunjang proses pengembangan aplikasi. Tugas seperti API Request dilakukan dengan *library* Ktor, serialisasi JSON dengan KotlinX Serializer, *data persistence* dengan SQLDelight, dan *logging* untuk kedua *platform* oleh *library* Napier.

KMM mendukung kaidah *Clean Architecture* dengan memisahkan komponen menjadi *data, domain,* dan *interactors*. Data mendefinisikan dari mana data akan diambil, secara lokal dari database ataupun dari *network* dengan melakukan API Request. SQLDelight merupakan *library* yang digunakan untuk proses *caching* sehingga beberapa data tetap tersedia meskipun *user* tidak memiliki jaringan internet yang aktif. Domain mendefinisikan berbagai macam model/ kelas yang digunakan pada aplikasi. Proses konversi dari JSON Response menjadi Kotlin Model didukung oleh *library KotlinX Serializer*. Seluruh *properties dan method* sebuah kelas di-definisikan pada bagian Domain. Interactors merupakan sebuah komponen yang mendefinisikan berbagai macam *use-case* seperti melakukan Insert, Select, Update, ataupun Delete dari aplikasi ke Database. Kedua *platform* baik dari Android maupun iOS akan berinteraksi dengan *business logic* melalui *interactor* yang telah dibuat. Presenter/ View dari setiap *platform* akan dibuat secara *native* menggunakan bahasa utama setiap *platform* yaitu Java/ Kotlin untuk Android dan Objective-C atau Swift/SwiftUI untuk iOS.



Gambar 3.2 Logo Kotlin Multiplatform Mobile

3.3 Xcode

Xcode adalah sebuah Integrated Development Environment (IDE) resmi untuk aplikasi macOS, iPadOS, watchOS, dan iOS dari Apple. Xcode digunakan sebagai *code editor* untuk memprogram aplikasi iOS dengan fitur-fitur umum yang ada pada IDE pada umumnya seperti *debugger*, *layout inspector*, dan lain-lain. Xcode juga menyediakan *emulator* dengan berbagai jenis model perangkat milik Apple agar pengembang dapat melakukan uji coba aplikasi yang dibuat pada berbagai macam perangkat.

Gambar 3.3 Logo Xcode

3.4 SwiftUI

SwiftUI adalah sebuah *declarative framework* yang menghadirkan sebuah cara penyusunan komponen tampilan secara deklaratif menggunakan bahasa pemrograman Swift. SwiftUI berfokus pada *state change* yaitu pembaharuan tampilan secara otomatis ketika terjadi perubahan data. SwiftUI juga menyediakan *event handler* untuk mendeteksi *taps*, *gestures*, dan berbagai macam input lainnya sehingga pengembang berfokus pada alur data, tampilan yang dipresentasikan, dan efek input dari *user*.

Gambar 3.4 Logo SwiftUI

3.5 Git

Git adalah sebuah *version control software* gratis dan *open-source* yang memudahkan pengembang untuk membuat *checkpoint* untuk file dalam proyek. Git umumnya digunakan oleh lebih dari satu pengembang untuk berkolaborasi dalam suatu *source code*. Git digunakan sebagai basis dari *cloud repository* umum yang ada seperti GitHub, GitLab, BitBucket, dan lainnya.

Gambar 3.5 Logo GIT

3.6 Gitlab

GitLab adalah sebuah *cloud repository* yang digunakan untuk menyimpan *source code* berbagai macam proyek dan digunakan lebih dari satu pengembang untuk berkolaborasi. GitLab juga mendukung Software Development Lifecycle (SDLC) dengan menyediakan berbagai macam fitur seperti *issue-tracking*, *continuous integration* dan *continuous depolyment* (CI/CD), dan fitur lainnya.

Gambar 3.6 Logo GitLab

3.7 Trello

Trello adalah sebuah *software* berbasis web yang menyediakan fitur untuk membuat Kanban Board sebagai sarana manajemen proyek. Trello menyediakan fitur untuk mengorganisir informasi seperti bagian apa yang ada pada sebuah proyek, siapa yang sedang mengerjakan, bagian yang telah selesai, dan bagian yang memerlukan perbaikan. Trello digunakan para pengembang untuk menyusun tugas berdasarkan priortias dengan memberi label umum seperti *blocker, blocked, urgent,* dan lainnya sesuai kebutuhan tim.

Gambar 3.7 Logo Trello