

BAB 2

LANDASAN TEORI

2.1 Manajemen Keuangan

Manajemen keuangan adalah proses yang melibatkan pengelolaan siklus keuangan secara menyeluruh, baik untuk sebuah organisasi maupun individu. Tahapan-tahapan dalam manajemen keuangan meliputi perencanaan, penganggaran, pelaksanaan, pengawasan, dan evaluasi keuangan. Ini mencakup pengambilan keputusan strategis terkait dengan penggunaan sumber daya keuangan, investasi, pengelolaan risiko keuangan, serta pemantauan kinerja keuangan secara keseluruhan [9].

2.2 Android Studio

Android Studio adalah lingkungan pengembangan terintegrasi (IDE) resmi yang digunakan untuk membuat aplikasi Android. Dikembangkan oleh Google, Android Studio menyediakan berbagai alat dan fitur yang memudahkan pengembang dalam merancang, menguji, dan menerapkan aplikasi Android [10]. Berikut adalah beberapa fitur utama Android Studio.

1. Sistem versi yang berbasis pada Gradle yang dapat disesuaikan.
2. Emulator yang memiliki kinerja cepat dan berbagai fitur.
3. Lingkungan pengembangan yang terintegrasi untuk semua perangkat Android.
4. Instant Run untuk mengaplikasikan perubahan ke aplikasi yang sedang berjalan tanpa perlu membuat APK baru.
5. Template kode dan integrasi dengan GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
6. Alat pengujian dan kerangka kerja yang luas.
7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan menangani masalah lainnya.
8. Dukungan untuk bahasa pemrograman C++ dan NDK.

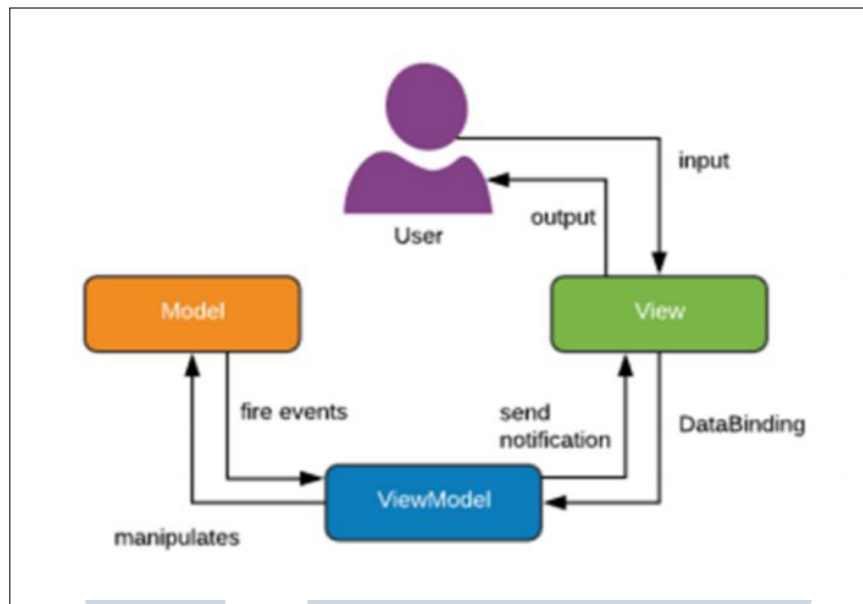
2.3 Kotlin

Kotlin, sebuah bahasa pemrograman yang berjalan di atas Java Virtual Machine (JVM), dirancang oleh JetBrains, perusahaan yang juga menciptakan IntelliJ IDEA. IntelliJ IDEA adalah Lingkungan Pengembangan Terpadu (IDE) untuk Java. Android Studio, IDE resmi untuk pembuatan aplikasi Android, dibangun di atas platform IntelliJ [7].

2.4 Arsitektur Model-View-ViewModel(MVVM)

MVVM adalah varian modern dari Model-View-Controller (MVC) dan tujuan intinya adalah memisahkan antara komponen Model dan komponen View. Komponen dalam MVVM dapat dijelaskan sebagai berikut.

1. *Model*, Komponen ini mewakili logika bisnis dan data. Hal tersebut berarti bahwa logika bisnis menentukan bagaimana agar data yang akan dimanipulasi terdapat di dalam komponen *Model*
2. *View*, Komponen ini mewakili komponen antarmuka dan pada dasarnya akan berisi komponen antarmuka, seperti layout XML , jetpack compose dsb.
3. *ViewModel*, merupakan komponen yang sangat esensial dalam arsitektur karena berperan dalam memisahkan tampilan, yaitu memastikan View terpisah dari *Model* dan pada saat yang sama, bertindak sebagai pengontrol yang memfasilitasi interaksi dan koordinasi antara *View* dan *Model*. *ViewModel* juga mencakup perintah dan metode yang membantu dalam mempertahankan keadaan tampilan serta membantu dalam memanipulasi *Model* sesuai dengan tindakan yang dilakukan pada *View*. Selain itu, *ViewModel* juga membantu dalam memicu peristiwa dalam komponen *View* itu sendiri [11].



Gambar 2.1. Model View ViewModel

2.5 Unified Model Language (UML)

Unified Modeling Language, yang lebih dikenal sebagai UML, merupakan salah satu metode dalam teknik rekayasa perangkat lunak yang digunakan untuk memvisualisasikan alur dan cara kerja sistem, termasuk fungsinya, tujuannya, dan mekanisme kontrol sistem tersebut. Dalam bidang analisis dan desain sistem informasi pada teknik rekayasa perangkat lunak, saat ini lebih banyak menggunakan kombinasi antara konsep pemrograman berorientasi objek dan teknik pengembangan perangkat lunak. Dalam pendekatan ini, suatu sistem dianggap sebagai objek independen yang sudah mencakup data dan proses, atau mampu berfungsi secara mandiri dalam satu set sistem [12].

2.6 User Acceptance Test (UAT)

UAT sebagai bagian integral dari siklus pengujian perangkat lunak, memiliki tujuan yang krusial dalam memastikan bahwa produk yang dikembangkan sesuai dengan preferensi dan kebutuhan para pengguna akhir. Proses pengumpulan data dalam pengujian UAT dilakukan melalui penggunaan kuesioner sebagai alat utama. Kuesioner dipilih karena dapat memberikan wawasan yang lebih menyeluruh, memungkinkan para responden untuk memberikan tanggapan secara lebih rinci, dan menciptakan lingkungan di mana mereka merasa lebih nyaman untuk menyampaikan pandangan mereka tanpa rasa takut bahwa jawaban mereka

tidak sesuai dengan ekspektasi yang diharapkan saat mengisi formulir pertanyaan [13].

2.7 Skala Likert

Skala Likert merupakan sebuah alat psikometrik yang umumnya digunakan dalam penyusunan angket atau kuesioner, seringkali digunakan dalam penelitian berupa survei. Dalam skala Likert, terdapat dua jenis pertanyaan, yaitu pertanyaan positif yang digunakan untuk mengukur aspek positif, dan pertanyaan negatif yang digunakan untuk mengukur aspek negatif. Pertanyaan positif diberi penilaian skor 5, 4, 3, 2, dan 1, sementara pertanyaan negatif diberi penilaian skor 1, 2, 3, 4, dan 5. Definisi lain menjelaskan bahwa skala ini digunakan untuk mengumpulkan data, baik yang bersifat kualitatif maupun kuantitatif, guna mengevaluasi atau mengukur aspek-aspek tertentu. Data yang dihasilkan dari skala ini digunakan untuk memahami pandangan, persepsi, dan penilaian individu terhadap suatu fenomena yang sedang diamati atau diteliti [14]. berikut rumus yang digunakan dalam skala likert [15], [16].

1. Menghitung masing-masing pertanyaan kuesioner.

$$\text{Skor Total} = (1 \times P1) + (2 \times P2) + (3 \times P3) + (4 \times P4) + (5 \times P5)$$

- P1 = Total responden menjawab "Sangat Tidak Setuju"
- P2 = Total responden menjawab "Tidak Setuju"
- P3 = Total responden menjawab "Netral"
- P4 = Total responden menjawab "Setuju"
- P5 = Total responden menjawab "Sangat Setuju"

Skor Total adalah hasil skor dari setiap aspek.

2. Menghitung skor yang dilakukan untuk setiap pertanyaan agar mendapatkan hasil perhitungan.

$$\text{Interpretasi (\%)} = \frac{\text{SkorTotal}}{Y} \times 100$$

Keterangan:

- Y= Skor maksimum setiap aspek
- Skor Total = Hasil skor setiap aspek

3. Menghitung Interval Skor.

Mengkalkulasi interval dan persentase penting untuk memahami evaluasi melalui teknik penentuan interval skor persentase.

$$I = \frac{50}{\text{Jumlah skor likert}}$$

Karena jumlah skor yang digunakan ada 5, maka perhitungan dapat dikerjakan.

$$I = \frac{50}{5}$$

$$I = 10$$

(10 merupakan jarak dari terendah 0% hingga tertinggi 100%)

Tabel 2.1. Kriteria Perhitungan skor

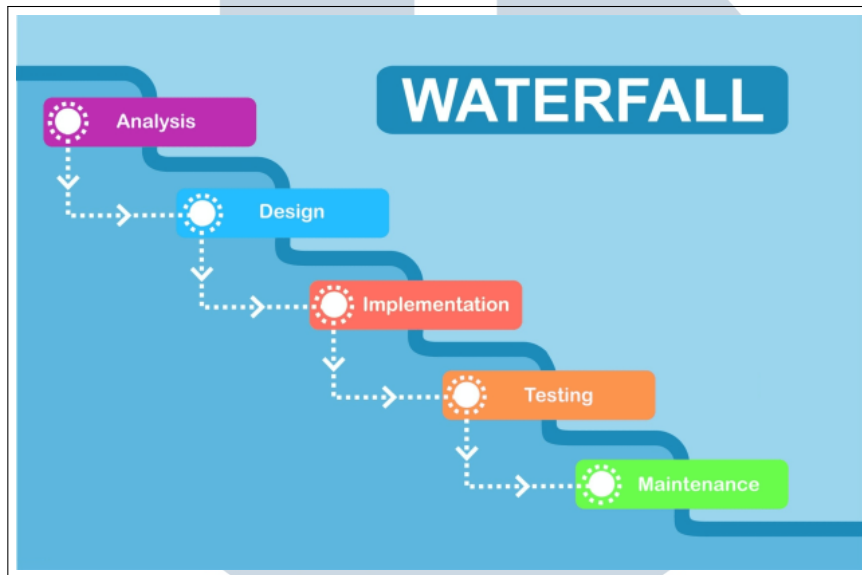
Keterangan	Persentase
Sangat Tidak Setuju	0% - 19,99%
Tidak Setuju	20% - 39,99%
Netral	40% - 59,99%
Setuju	60% - 79,99%
Sangat Setuju	80% - 100%

2.8 Waterfall

Model pengembangan perangkat lunak *Waterfall* merupakan salah satu metode yang sudah lama digunakan dalam rekayasa perangkat lunak. Pendekatan ini menyoroti urutan tahapan yang berjalan secara berurutan dan *linear* selama proses pengembangan. Berikut adalah penjelasan ringkas tentang model *Waterfall* [17].

1. Analisis: Langkah pertama dalam model Waterfall adalah melakukan analisis terhadap kebutuhan. Pada tahap ini, kebutuhan bisnis dan kebutuhan pengguna diidentifikasi secara detail untuk memastikan pemahaman yang jelas. Tujuannya adalah untuk menetapkan tujuan yang harus dicapai oleh perangkat lunak.
2. Perancangan: Setelah kebutuhan dipahami, langkah berikutnya adalah merancang sistem. Pada tahap ini, struktur keseluruhan sistem dan detail desainnya didefinisikan. Ini termasuk perencanaan fungsi, desain teknis, dan antarmuka pengguna.
3. Implementasi: Tahap selanjutnya adalah implementasi, di mana pengembang mulai menulis kode perangkat lunak sesuai dengan spesifikasi dan desain yang telah disepakati. Proses ini bertujuan untuk menghasilkan kode yang sesuai dengan rancangan yang telah dibuat sebelumnya.
4. Pengujian: Setelah implementasi selesai, perangkat lunak diuji secara menyeluruh untuk memastikan kinerjanya sesuai dengan kebutuhan yang telah ditetapkan. Pengujian dilakukan untuk mendeteksi dan memperbaiki bug, serta memastikan bahwa perangkat lunak beroperasi dengan baik dalam berbagai situasi.
5. Pemeliharaan: Tahap terakhir adalah pemeliharaan, di mana perangkat lunak terus diperbaiki dan ditingkatkan berdasarkan umpan balik dari pengguna dan perubahan kebutuhan bisnis.

Dalam model *Waterfall*, setiap tahapan harus diselesaikan secara lengkap sebelum memulai tahapan berikutnya, mirip dengan air yang mengalir turun dari atas ke bawah dalam air terjun. Metode *waterfall* yang digunakan pada penelitian dapat dilihat pada gambar 2.2 .



Gambar 2.2. Metode Waterfall

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA