

## BAB II

### LANDASAN TEORI

Pada bab ini berisikan mengenai pembahasan teori pengembangan *system* dengan menggunakan Android studio, teori penelitian terkait dengan perancangan *system* informasi jual beli bahan baku berbasis *mobile*, teori mengenai *design system* menggunakan *UML*, konsep *MVC*, teori mengenai database *Firebase*.

#### 2.1 Tinjauan Pustaka

##### 2.1.1 UML (Unified Modelling Language)

*Unified Modeling Language (UML)* adalah sebuah bahasa pemodelan grafis yang digunakan sebagai standar untuk memodelkan *system* dengan metodologi pemodelan berorientasi objek.

*Unified Modeling Language (UML)* di standarkan oleh *Object Management Group (OMG)*. *UML* pertamakali dipopulerkan oleh Grady Booch dan James Rumbaugh pada akhir tahun 1994. Kemudian Ivar Jacobos yang merupakan pimpinan dari *Object Oriented Development* bergabung. Pada tahun 1996 *UML* mulai digunakan sebagai *tools* untuk memodelkan *system* pada IBM dan i-logix [1].

Berikut adalah 4 (empat) fungsi dan kegunaan dari *UML* yaitu:

1. *Virtualizing*
2. *Specifying*
3. *Constructing*
4. *Documenting*

*Virtualizing* adalah sebagai alat komunikasi atau konsep model antara tim pengembang *system* ( *system* analis dengan programmer).

*Specifying* adalah sebagai *tools* yang digunakan untuk memodelkan *system* secara tepat dan jelas.

*Constructing* adalah *UML* sebagai bahasa grafis (bukan bahasa pemograman) mampu melakukan *mapping* dari konsep model kedalam bahasa pemograman.

*Documenting* adalah *UML* digunakan sebagai *tools* untuk melakukan dokumentasi teknis sebuah *system*.

Diagram-diagram yang terdapat pada UML sebagai berikut:

1. *Use Case Model*
2. *Class Diagram*
3. *Object Diagram*
4. *Activity Diagram*
5. *Sequence Diagram*
6. *Collaboration Diagram*
7. *Statechart Diagram*
8. *Component Diagram*
9. *Deployment Diagram*

Berikut adalah penjelasan dari jenis-jenis pemodelan *UML* diagram diatas yaitu:

1. *Use Case Model*

Use case model merupakan kumpulan diagram dan text yang saling bekerja sama untuk mendokumentasikan bagaimana user atau actor berinteraksi dengan system, usecase model terdiri dari beberapa diagram yaitu:

1. *Use case diagram* adalah untuk menjelaskan konsep hubungan antara system dengan dunia luar, atau menggambarkan dan merepresentasikan actor, *use case*, dan dependencies.
2. *Use case narrative* adalah deskripsi yang menjelaskan *use case diagram*.
3. *Use case secenario* adalah pemecahan kemungkinan logika pada *use case diagram*.

4. *Class Diagram*

Class diagram adalah diagram yang digunakan untuk merepresentasikan kelas, komponen-komponen kelas dan hubungan masing-masing kelas, mendeskripsikan jenis-jenis objek dalam sistem.

5. *Object Diagram*

*Object diagram* adalah diagram yang digunakan untuk menggambarkan dan merepresentasikan objek dan hubungan antar objek tersebut.

#### 6. *Activity Diagram*

*Activity diagram* adalah diagram *UML* yang digunakan untuk menggambarkan alur aktivitas dari suatu proses. *Activity diagram* ini sangat mirip dengan *flowchart*.

#### 7. *Sequence Diagram*

*Sequence diagram* adalah diagram yang menggambarkan interaksi antar objek. *Sequence diagram* secara khusus menjabarkan *behavior* sebuah scenario tunggal.

#### 8. *Collaboration Diagram*

*Collaboration diagram* hampir sama dengan *sequence diagram*, dapat dikatakan *collaboration diagram* merupakan pengganti *sequence diagram*.

#### 9. *Statechart Diagram*

*Statechart Diagram* adalah teknik umum yang digunakan untuk menggambarkan *behavior* sebuah *system*.

#### 10. *Component Diagram*

*Component diagram* adalah diagram yang menggambarkan dan merepresentasikan model fisik dari implementasi *system software*.

#### 11. *Deployment Diagram*

*Deployment diagram* adalah diagram yang merepresentasikan model fisik dari hardware serta integrasi dan distribusi software pada arsitektur hardware tersebut.

### 2.1.2 Model Waterfall

Metode *Waterfall* adalah salah satu metode dalam SDLC yang mempunyai ciri khas pengerjaan yaitu setiap fase dalam *waterfall* harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya. Artinya focus terhadap masing-masing fase dapat dilakukan maksimal karena jarang adanya pengerjaan yang sifatnya parallel walaupun dapat saja terjadi pararelisme dalam *waterfall* [5]. Fase dalam metode waterfall, tahapan-tahapan dari metode waterfall adalah sebagai berikut:

1. Requirement Analysis

Seluruh kebutuhan software harus bisa didapatkan dalam fase ini termasuk didalamnya kegunaan software yang diharapkan pengguna dan batasan software. Informasi ini biasanya dapat diperoleh melalui wawancara, survei atau diskusi.

2. System Design

Tahap ini dilakukan sebelum melakukan ke proses coding atau pemrograman. Tahap ini bertujuan untuk membuat design dari hasil analisis yang telah didapat sesuai dengan kebutuhan hardware dan system secara keseluruhan.

3. Implementation

Ditahap ini dilakukannya proses coding atau pemrograman. Pembuatan software dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya.

4. Integration and Testing

Ditahap ini dilakukannya penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah software yang dibuat telah sesuai dengan design yang telah dibuat sebelumnya.

5. Operation and Maintenance

Tahap ini adalah proses dilakukannya pemeliharaan, seperti kemungkinan adanya bugs, error, dan malfungsi terhadap software aplikasi yang telah dibuat.

### 2.1.3 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

- Sistem versi berbasis Gradle yang fleksibel
- Emulator yang cepat dan kaya fitur
- Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
- Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
- Alat pengujian dan kerangka kerja yang ekstensif
- Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
- Dukungan C++ dan NDK
- Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

Laman ini berisi pengantar dasar fitur-fitur Android Studio. Untuk memperoleh rangkuman perubahan terbaru, lihat Catatan Rilis Android Studio [3].

#### 2.1.4 Struktur Proyek

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup:

- Modul aplikasi Android
- Modul Pustaka
- Modul Google App Engine

Secara default, Android Studio akan menampilkan file proyek Anda dalam tampilan proyek Android, seperti yang ditampilkan dalam gambar 1. Tampilan disusun berdasarkan modul untuk memberikan akses cepat ke file sumber utama proyek Anda.

Semua file versi terlihat di bagian atas di bawah **Gradle Scripts** dan masing-masing modul aplikasi berisi folder berikut:

- **manifests:** Berisi file **AndroidManifest.xml**.
- **java:** Berisi file kode sumber Java, termasuk kode pengujian JUnit.
- **res:** Berisi semua sumber daya bukan kode, seperti tata letak XML, string UI, dan gambar bitmap.

Struktur proyek Android pada disk berbeda dari representasi rata ini. Untuk melihat struktur file sebenarnya dari proyek ini, pilih **Project** dari menu tarik turun **Project** (dalam gambar 1, struktur ditampilkan sebagai **Android**) [3].

#### 2.1.5 Firebase

Firebase adalah BaaS (Backend as a Service) yang saat ini dimiliki oleh Google. Firebase ini merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan Mobile Apps Developer. Dengan adanya Firebase, apps developer bisa fokus mengembangkan aplikasi tanpa harus memberikan effort yang besar untuk urusan backend [2]. Dengan pengertian di atas penulis akan menggunakan

firebase untuk menghubungkan data-data informasi dari aplikasi yang di buat dan di hubungkan ke firebase.

Berikut adalah beberapa fitur dari firebase yang digunakan untuk membuat aplikasi swasembada dan penjelasannya :

- Firebase authentication.

Sebagian besar aplikasi perlu mengetahui identitas pengguna. Dengan mengetahui identitas pengguna, aplikasi dapat menyimpan data pengguna secara aman di cloud dan memberikan pengalaman personal yang sama di setiap perangkat pengguna [8].

- Cloud storage.

*Cloud Storage for Firebase* adalah layanan penyimpanan objek yang andal, sederhana, dan hemat biaya yang dibuat untuk skala *Google*. *Firestore* SDK untuk *Cloud Storage* menambahkan keamanan *Google* pada *upload* dan *download file* untuk aplikasi yang akan di buat [9].

- Firebase realtime database.

*Firebase Realtime Database* adalah *database* yang di-host di cloud. Data disimpan sebagai *JSON* dan disinkronkan secara realtime ke setiap klien yang terhubung, dengan menggunakan ini semua klien akan berbagi sebuah *instance realtime database* dan menerima *update* data terbaru secara otomatis [10].

STT - NF

## 2.2 Penelitian Terkait

Berdasarkan studi pustaka yang telah dikerjakan, telah ditemukan beberapa penelitian yang dijadikan acuan dalam pengembangan penelitian yang akan di buat, diantaranya adalah sebagai berikut :

Table 1 : penelitian terkait aplikasi swasembada

No	Aplikasi penelitian terkait	Aplikasi WEB	Aplikasi MOBILE	Transaksi cash	Transaksi Debit / bank transfer
1	Limakilo (2015)	Memiliki aplikasi web, design sangat bagus dan mudah di mengerti, bias transaksi di aplikasi web limakilo.	Tidak memiliki Aplikasi mobile limakilo belum ada info mengenai aplikasi mobile.	Tidak tersedia	Tersedia
2	Eragano (2016)	Memiliki aplikasi web, design menarik, aplikasi web ini hanya menampilkan detail perusahaan dan produk apa yang ditawarkan, tidak ada transaksi penjualan di dalam web.	Memiliki aplikasi mobile tetapi masih rumit untuk menggunakan aplikasi ini, dan sulit dalam melakukan login dan registrasi. Dan belum ada update terbaru untuk aplikasi ini.	Tidak tersedia	Tersedia
3	TaniHub (2016)	Memiliki aplikasi web, design nya sangat bagus mudah	Tidak memiliki aplikasi mobile, melainkan	Tidak tersedia	Tersedia

		dimengerti,terdapat transaksi penjualan tetapi menghubungkan ke aplikasi lain.	mengkaitkan dengan aplikasi tokopedia dalam penjalannya.		
4	Swasembada (2017)	Memiliki aplikasi web, aplikasi ini sedang di kembangkan, memiliki design, transaksi penjual dan pembeli antara petani dan pembeli masyarakat local.	Memiliki aplikasi mobile dan sedang di kembangkan dari system penjualan dan pembelian antara petani dan pembeli.	Tersedia Metode bayar di tempat	Tersedia

STT - NF