

BAB II

TINJAUAN PUSTAKA

Bab ini membahas tentang profil singkat PT. Duta Semesta Utama dan anak perusahaannya, PT. Indomedia Mega Corpora serta membahas mengenai *Mobile Apps Prototyping* dan *tools* yang digunakan.

2.1 Profil PT. Duta Semesta Utama

PT. Duta Semesta Utama resmi didirikan oleh Sulistyo Wasono pada tahun 2014. PT. Duta Semesta Utama adalah sebuah perusahaan yang bergerak di bidang *Real Estate Development*, arsitektur, kontraktor, dan juga sebagai pemasok umum. PT. Duta Semesta Utama memiliki motto “Jujur, Amanah, Komunikatif, Visi dan Inovasi, Peduli”. Perusahaan ini beralamat di Jl. Citra Garden Waterfront E7, Kota Sidoarjo, Jawa Timur.

PT. Duta Semesta Utama memiliki sebuah anak perusahaan yang bernama PT. Indomedia Mega Corpora yang bergerak di bidang pembuatan perangkat lunak/softwarehouse. Anak perusahaan ini berada di alamat yang sama dengan PT. Duta Semesta Utama, yaitu Jl. Citra Garden Waterfront E7, Kota Sidoarjo, Jawa Timur. Saat ini PT. Indomedia Mega Corpora sedang membuat aplikasi *mobile* tentang pemesanan travel umrah bernama “Hijrahku”.

2.1.1 Visi dan Misi

a. Visi

Menjadi perusahaan IPO 2020

b. Misi

1. Meningkatkan kualitas semua lini bisnis dari anak perusahaan
2. Ekspansi di segala faktor perusahaan
3. Membangun jaringan bisnis di segala segmen

2.1.2 Struktur Organisasi

Direktur Utama : Sulistyo Wasono
Direktur Marketing : Edwib Susilo, S.E., M.M.
Direktur Keuangan : Dwi Handayani, S.E.
Direktur Operasional : Bambang Gunawan Winardi
Direktur HRD : Stevani Ersantanti, S. Psi.

2.2 Aplikasi *Mobile*

Pengertian aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem computer (Irawan, 2009). *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat lain. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel, seperti pager, seperti telepon seluler (Mulyadi, 2010).

2.3 Prototype

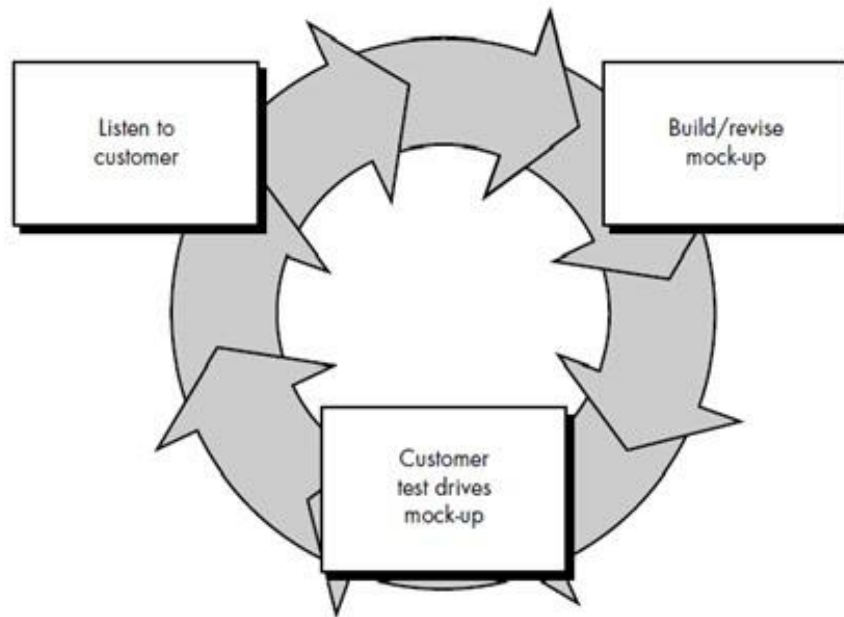
Prototype adalah satu versi dari sebuah sistem potensial yang memberikan ide bagi para pengembang dan calon pengguna, bagaimana sistem akan berfungsi dalam bentuk yang telah selesai (Darmawan, 2013).

Seperti halnya ketika akan menilai sebuah mobil tanpa mencobanya, *user* juga tidak dapat menilai dari spesifikasi fungsional, bagaimana sistem akan berlaku dan berjalan. Tetapi jika *user* dapat melihat, menyentuh dan menggunakan ‘model’ atau *prototype* dari tujuan sistem dapat langsung menilai kegunaan sistem. Jika perubahan diperlukan, *prototype* dapat dimodifikasi, memungkinkan dimodifikasi beberapa kali sampai keadaan yang ditetapkan *user*.

Prototype didefinisikan sebagai alat yang memberikan ide bagi pembuat maupun pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya, dan proses untuk menghasilkan sebuah *prototype* disebut *prototyping* (McLeod, Raymond, & Schell, 2008).

Prototyping adalah proses pembuatan model sederhana *software* yang mengizinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototyping* memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan dibuat.

2.3.1 Tahapan - tahapan *Prototyping*



Gambar 2.1 *Prototype* model menurut (Pressman, 2002).

Tahap-tahap pengembangan *prototype* model menurut (Pressman, 2002) adalah:

a) Mendengarkan pelanggan

Pada tahap ini dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar keluhan dari pelanggan. Untuk membuat suatu sistem yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana sistem yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

b) Merancang dan Membuat *prototype*

Pada tahap ini, dilakukan perancangan dan pembuatan *prototype sistem*. *Prototype* yang dibuat disesuaikan dengan kebutuhan sistem yang telah didefinisikan sebelumnya.

c) Uji Coba

Pada tahap ini, *prototype* dari sistem diuji coba oleh pengguna. Kemudian dilakukan evaluasi kekurangan - kekurangan dari kebutuhan pelanggan. Pengembang kemudian kembali mendengar keluhan dari pelanggan untuk memperbaiki *prototype* yang ada.

2.4 Konsep Dasar UML

UML singkatan dari *Unified Modeling Languages* yang berarti bahasa pemodelan standar yang memiliki sintak dan semantik (Widodo, 2011).

Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram tetapi juga menceritakan konteksnya. UML dapat diaplikasikan untuk maksud tertentu, antara lain untuk:

- a) Merancang perangkat lunak
- b) Sarana komunikasi antara perangkat lunak dalam proses bisnis.
- c) Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
- d) Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

2.4.1 Jenis - jenis Diagram UML

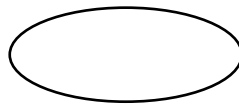
UML menyediakan 4 macam diagram untuk memodelkan aplikasi perangkat lunak berorientasi objek. Yaitu:

a. *Use Case Diagram*

Use Case Diagram adalah sebuah diagram yang menunjukkan hubungan antara *actors* dan *use cases*. Digunakan untuk analisis dan desain sebuah *system* (Ambler, 2005). *Use Case Diagram* memiliki bagian-bagian sebagai berikut:

1. *Use cases*

Use cases menjelaskan tentang tindakan/aksi yang dilakukan oleh *actors*. *Use case* digambarkan dalam bentuk *elips* yang horizontal.



Gambar 2.2 Gambaran *use case*.

2. *Actors*

Actors adalah seorang peran yang berinteraksi dengan *system*. *Actors* meliputi baik manusia maupun organisasi yang saling bertukar informasi.

3. *Relationship*

Relationship adalah hubungan antara *use cases* dengan *actors*. *Relationship* dalam *use case diagram* meliputi asosiasi antara *actor* dan

use case, asosiasi antara 2 *use case*, generalisasi antara 2 *actor*, dan generalisasi antara 2 *use case*.

b. *Class Diagram*

Class Diagram adalah sebuah diagram yang menunjukkan hubungan antar *class* yang di dalamnya terdapat atribut dan fungsi dari suatu objek (Ambler, 2005). Dalam penggunaannya, *Class Diagram* mempunyai 3 relasi yaitu:

1. *Association*

Association adalah sebuah hubungan yang menunjukkan adanya interaksi antar *class*. Hubungan ini dapat ditunjukkan dengan garis dengan mata panah terbuka di ujungnya yang mengindikasikan adanya aliran pesan dalam satu arah.

2. *Generalization*

Generalization adalah sebuah hubungan antar *class* yang bersifat dari khusus ke umum.

3. *Constraint*

Constraint adalah sebuah hubungan yang digunakan dalam sistem untuk memberi batasan pada sistem sehingga didapat aspek yang tidak fungsional.

c. *Activity Diagram*

Activity Diagram adalah suatu diagram yang menggambarkan konsep aliran data/kontrol, aksi terstruktur serta dirancang dengan baik dalam suatu sistem.

d. *Sequence Diagram*

Sequence Diagram adalah sebuah diagram yang menggambarkan kolaborasi dari objek-objek yang saling berinteraksi antar elemen dari suatu *class*. Komponen-komponen dalam *Sequence Diagram* antara lain:

1. *Activations*

Activations menjelaskan tentang eksekusi dari fungsi yang dimiliki oleh suatu objek.

2. *Actor*

Actor menjelaskan tentang peran yang melakukan serangkaian aksi dalam suatu proses.

3. *Collaboration boundary*

Collaboration boundary menjelaskan tentang tempat untuk lingkungan percobaan dan digunakan untuk memonitor objek.

4. *Parallel vertical lines*

Parallel vertical lines menjelaskan tentang suatu garis proses yang menunjuk pada suatu *state*.

5. *Processes*

Processes menjelaskan tentang tindakan/aksi yang dilakukan oleh aktor dalam suatu waktu.

6. *Window*

Window menjelaskan tentang halaman yang sedang ditampilkan dalam suatu proses.

7. *Loop*

Loop menjelaskan tentang model logika yang berpotensi untuk diulang beberapa kali.

2.5 Aplikasi yang Dipakai

2.5.1 Android Studio

Android Studio adalah *Integrated Development Enviroment* (IDE) untuk sistem operasi Android, yang dibangun di atas perangkat lunak *JetBrains IntelliJ IDEA* dan didesain khusus untuk pengembangan Android. IDE ini merupakan pengganti dari *Eclipse Android Development Tools* (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi android.

Android studio sendiri pertama kali diumumkan di *Google I/O conference* pada tanggal 16 Mei 2013. Ini merupakan tahap *preview* dari versi 0.1 pada Mei 2013, dan memasuki tahap beta sejak versi 0.8 dan mulai dirilis pada Juni 2014.

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, dan modul *Google App Engine*.

2.5.2 SQLite Database

SQLite adalah sebuah *software RDBMS* (*Relational Database Management Sistem*) yang mendukung secara *native* (asli) untuk perangkat Android. SQLite merupakan suatu sistem manajemen *database* yang mempunyai sifat ACID compliant yang diprogram dengan bahasa C, dan mempunyai *size* atau ukuran

memori yang relatif kecil. Karena SQLite termasuk *database engine* yang *embedded* (tersemat), jadi perintah SQLite yang bisa digunakan hanya perintah-perintah standar saja. Serta SQLite hanya mendukung tipe data seperti NUMERIC, DATETIME, TEXT, dan beberapa tipe data lainnya.

Android memiliki fasilitas untuk membuat *database* yang dikenal dengan SQLite, SQLite merupakan salah satu software yang *embedded*, kombinasi SQL *interface* dan penggunaan *memory* yang sangat sedikit dengan kecepatan sangat tinggi. SQLite di android termasuk dalam *Android runtime*, sehingga setiap versi dari android dapat membuat *database* dengan SQLite. Dalam sistem android memiliki beberapa teknik untuk melakukan penyimpanan data. Teknik yang umum digunakan adalah sebagai berikut:

- a) *Shared Prefences*, yaitu menyimpan data beberapa nilai (*value*) dalam bentuk *groups key* yang dikenal dengan *prefences*.
- b) *Files*, yaitu menyimpan data dalam *file*, dapat berupa menulis ke *file* atau membaca dari *file*.
- c) *SQLite Databases*, yaitu menyimpan data dalam bentuk *Database*.
- d) *Content Providers*, yaitu menyimpan data dalam bentuk *content providers service*.

2.5.3 Adobe XD

Tahun 2016 Adobe memperkenalkan Adobe XD atau Adobe *Experience Design* sebagai perangkat lunak pertama yang memiliki fungsi lengkap untuk mendesain dan membuat *prototype* laman dan aplikasi gawai. Kedua aktivitas pengembangan *prototype* produk UX dan UI ini dapat dilakukan dalam satu perangkat lunak. Perangkat lunak ini lebih menekankan pada desain dan pengembangan produk digital yang interaktif dengan lebih praktis dan ringkas.