

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1. Tinjauan Pustaka

Penelitian tentang aplikasi *mobile* berbasis android yang dibuat oleh universitas atau berisi info seputar kampus atau panduan bagi mahasiswa atau terkait kalangan akademisi atau perguruan tinggi yang ada relevansinya dengan penelitian yang hendak dilakukan ada beberapa, pertama adalah penelitian Panduan Kegiatan Akademik Universitas Gunadarma yang dilakukan Fahrul (2012) dari Universitas Gunadarma Jakarta berupa panduan atau info kegiatan akademik dan fasilitas kampus dengan menggunakan Bahasa Indonesia sebagai bahasa antarmuka dan Eclipse Galileo, SQLite Expert Personal Edition V3.4 sebagai bahasa pemrogramannya, mengenai tampilan didominasi tulisan daripada gambar.

Penelitian kedua yang ditinjau adalah Sistem Informasi Akademik (SIA) STMIK GI MD Palembang versi android yang dibuat kolektif oleh Yazid A., Abdul H.A. dan M. Rachmadi (2013), isinya menyampaikan notifikasi atau pemberitahuan dari sistem tersebut kepada mahasiswa dan dosen berupa materi perkuliahan, info nilai, jadwal, tugas dan pengumuman dilengkapi dengan fitur diskusi. Aplikasi ini menggunakan Bahasa Indonesia sebagai bahasa antarmuka dan bahasa pemrogramannya menggunakan Android SDK, Eclipse, ADT, Java, MySQL, ulasan mengenai tampilannya sudah simpe, sistematis dan ada fitur notifikasi.

Sementara penelitian ketiga yang dilakukan oleh Giustia Puspa Geoda (2014) berupa Sistem Informasi Geografis (SIG) yang memetakan kampus dan lokasi

gedung di Universitas Diponegoro Semarang dengan memanfaatkan fasilitas dan fitur Google Maps, dan menampilkan data dosen, serta info fakultas dan jurusan. Bahasa antarmuka juga menggunakan Bahasa Indonesia, sedang bahasa pemrogramannya menggunakan Android SDK, Eclipse, ADT, Java, dan MySQL. Aplikasi ini cukup simpel menunya, fitur sederhana tapi tampilannya *eye catchy*.

Selanjutnya penelitian keempat tentang Pendaftaran Siswa Baru secara Online SMP Nurul Huda Banyuputih Situbondo Berbasis Android yang dibuat oleh Wahyudianto (2013), mahasiswa STMIK PGRI Banyuwangi dengan bahasa antarmuka Bahasa Indonesia dan menggunakan bahasa pemrograman JSON, XML dan IDE Eclipse Galileo 3.5, menu, fitur dan tampilannya sangat sederhana hanya menampilkan data pelamar dan calon siswa yang lulus dan tidak lulus.

Penelitian yang terakhir ditinjau adalah pustaka dari STMIK AKAKOM yang berjudul Aplikasi Tuntunan Pemakaman Jenazah Berbasis Android dibuat oleh Zainal Abidin (2014) yang menggunakan bahasa antarmuka Bahasa Indonesia dan Bahasa Inggris dan bahasa pemrogramannya Java dengan editor Eclipse dan SQLite sebagai databasenya. Kontennya berisi cara-cara prosesi pemakaman kaum muslimin dari pemandian, pengkhafanan, shalat jenazah hingga penguburan jenazah. Secara tampilan masih menggunakan animasi 2D, akan lebih menarik kalau menggunakan animasi 3D dan ada fitur resolution auto size untuk smartphone yang memiliki resolusi layar lebih besar.

Berdasarkan tinjauan pustaka terhadap kelima aplikasi tersebut ada beberapa perbedaan dan bahkan peningkatan untuk aplikasi yang akan penulis buat, berupa

fitur yang tidak ada di aplikasi-aplikasi tersebut atau kompilasi keunggulan semua aplikasi, yaitu:

1. konten berbahasa Inggris yang tidak ada di aplikasi-aplikasi tersebut;
2. konten tentang ijin belajar dan ijin tinggal (*staying legally*) di Indonesia belum ada di aplikasi-aplikasi tersebut;
3. menu akan dibuat lebih bertahap untuk memudahkan pengguna;
4. menghindari *long page*.

## **2.2. Dasar Teori**

Berikut adalah uraian, penjelasan, definisi, pengertian dasar dan istilah, serta ulasan yang didapat dari berbagai sumber atau referensi yang telah dipublikasikan dalam media cetak maupun elektronik (buku teks, artikel, dan lain-lain) yang digunakan dalam penelitian ini:

### **2.2.1. Admisi (*Admission*)**

Menurut KBBI adalah izin masuk, dalam konteks admisi suatu perguruan tinggi dimaksud sebagai izin masuk yang diberikan suatu perguruan tinggi kepada calon mahasiswa yang telah berhasil memenuhi serangkaian prosedur dan melengkapi persyaratan yang diperlukan.

### **2.2.2. Smartphone**

Ponsel cerdas (bahasa Inggris: *smartphone*) adalah telepon genggam yang mempunyai kemampuan dengan penggunaan dan fungsi yang menyerupai komputer. Belum ada standar pabrik yang menentukan arti ponsel cerdas. Bagi beberapa orang, ponsel cerdas merupakan telepon yang bekerja menggunakan seluruh

perangkat lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. Bagi yang lainnya, ponsel cerdas hanyalah merupakan sebuah telepon yang menyajikan fitur canggih seperti surel (surat elektronik), internet dan kemampuan membaca buku elektronik (*e-book*) atau terdapat papan ketik (baik sebagaimana jadi maupun terhubung keluar) dan penyambung VGA. Dengan kata lain, ponsel cerdas merupakan komputer kecil yang mempunyai kemampuan sebuah telepon. Sistem operasi yang ada pada smartphone, seperti: Windows Phone (Microsoft), WebOs (Berbasis Linux), Symbian (Nokia), iOS (Apple), Android (Google), Blackberry (RIM), Maemo (Nokia), Java, Nucleus, Palm OS. (Hilaliyah, 2017)

### **2.2.3. Mobile Application**

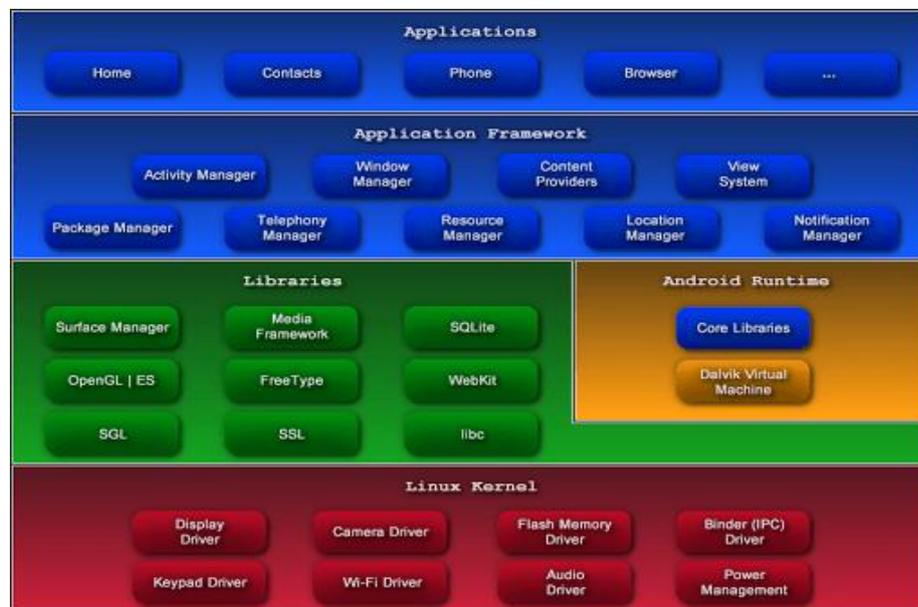
*Mobile Application* adalah sebuah aplikasi yang memungkinkan manusia melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler (*handphone*). Dengan menggunakan aplikasi *mobile*, manusia dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, *browsing*, *chatting*, e-mail, dan sebagainya. (Yonarisa, 2012)

### **2.2.4. Android**

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance,

konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. (UTY, 2013)

Gambar 2.1. di bawah merupakan susunan dari arsitektur pada android yang terdiri dari Linux Kernel. Kernel linux menyediakan driver layar, kamera, keypad, WiFi, Flash Memory, audio, dan IPC (Interprocess Communication) untuk mengatur aplikasi dan lubang keamanan:



Gambar 2.1. Arsitektur Platform Android

*Application Framework* yaitu kerangka aplikasi yang menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan *user interface* dan sumber daya aplikasi.

*Applications* yang merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan

ini berjalan dalam Android *runtime* dengan menggunakan kelas dan *service* yang tersedia pada *framework* aplikasi.

### **2.2.5. Android SDK**

Android SDK merupakan tools bagi para *programmer* yang ingin mengembangkan aplikasi berbasis google android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari debugger, libraries, handset emulator, dokumentasi, contoh kode, dan tutorial. (Onserda, 2013)

### **2.2.6. Java**

Java merupakan suatu teknologi perangkat lunak yang di dalamnya mencakup bahasa pemrograman, selain itu Java juga merupakan suatu *platform* yang memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan suatu program. Java sendiri pada awalnya dikembangkan pada lingkungan komputer oleh Sun Microsystem Inc. dengan tujuan untuk menghasilkan suatu bahasa komputer sederhana tanpa harus terikat pada arsitektur tertentu, maka tak heran jika dalam perkembangannya Java dapat dijalankan pada berbagai platform sistem operasi seperti Linux, Windows maupun Unix. Java pun semakin banyak dikenal karena keunikan dan keandalannya.

### **2.2.7.JDK (Java Development Kit)**

JDK adalah Perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java ke *bytecode* yang dapat dimengerti dan dapat dijalankan oleh JRE (Java Runtime Environment). JDK wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi berbasis java, namun tidak wajib terinstall di komputer yang akan menjalankan aplikasi yang dibangun dengan java.

### **2.2.8.IDE (Integrated Development Environment)**

IDE (Integrated Development Environment) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Sebuah IDE, atau secara bebas dapat diterjemahkan sebagai Lingkungan Pengembangan Terpadu, setidaknya memiliki fasilitas:

1. Editor, yaitu fasilitas untuk menuliskan kode sumber dari perangkat lunak;
2. Compiler, yaitu fasilitas untuk mengecek sintaks dari kode sumber kemudian mengubah dalam bentuk binari yang sesuai dengan bahasa mesin;
3. Linker, yaitu fasilitas untuk menyatukan data binari yang beberapa kode sumber yang dihasilkan compiler sehingga data-data binari tersebut menjadi satu kesatuan dan menjadi suatu program komputer yang siap dieksekusi;
4. Debugger, yaitu fasilitas untuk mengetes jalannya program, untuk mencari *bug*/kesalahan yang terdapat dalam program.

### **2.2.9.Android Studio**

Android Studio adalah sebuah IDE untuk pengembangan aplikasi di platform Android, sama seperti kombinasi antara Eclipse dan Android Developer Tools

(ADT), Android Studio juga dapat di-*download* di situs resmi Android: <http://developer.android.com/sdk/installing/studio.html>.

Android Studio menyediakan alat pengembang terintegrasi untuk pengembangan dan *debugging* dengan fitur-fitur unggulan: berbasis Gradle; android-spesifik refactoring dan perbaikan yang cepat; alat Lint untuk menangkap kinerja, kegunaan, versi kompatibilitas dan masalah lainnya; proGuard dan app-signature; wizard untuk design dan membuat komponen-komponen umum sebuah *layout editor* yang memungkinkan untuk *drag-and-drop* komponen UI, pratinjau layout pada beberapa konfigurasi layar, dan sebagainya; built-in dukungan untuk Google Cloud platform, sehingga mudah untuk mengintegrasikan Google Cloud Messaging dan App Engine sebagai komponen server-side.

#### **2.2.10.XML (Extensible Markup Language)**

XML (Extensible Markup Language) adalah bahasa *markup* untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen *markup* keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (HyperText Markup Language) yang merupakan bahasa standar untuk melacak Internet.

XML didesain untuk mampu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi. XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen, secara otomatis XML dapat digunakan untuk menggambarkan sembarang *view database*, tetapi dengan suatu cara yang standar.

Keunggulan XML bisa diringkas sebagai berikut: pintar (*intelligence*); dapat menangani berbagai tingkat (level) kompleksitas; dapat beradaptasi; dapat mengadaptasi untuk membuat bahasa sendiri, seperti Microsoft membuat bahasa MSXML atau Macromedia mengembangkan MXML; mudah pemeliharaannya; lebih sederhana; dan mudah dipindah-pindahkan (*portability*). (UNSRI, 2006)

### **2.2.11.UML (Unified Modeling Language)**

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yaitu Unified Modeling Language (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Sehingga penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

UML memiliki diagram grafis yang diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analisa atau rekayasa, ada 10 macam diagram untuk memodelkan aplikasi berorientasi objek, namun hanya dijelaskan empat macam diagram yang digunakan dalam aplikasi ini, yaitu:

#### **Use case diagram**

Use case Diagram digunakan untuk memodelkan semua bisnis proses berdasarkan perspektif pengguna sistem. Use case Diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan

mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.

### **Sequence Diagram**

Sequence Diagram merupakan salah satu Interaction Diagram yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu, obyek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.

### **Activity Diagram**

Activity Diagram adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

### **Class Diagram**

Class Diagram menggambarkan struktur statis *class* di dalam sistem, *class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *association* (terhubung satu sama lain), *aggregation* (salah satu *class* merupakan bagian dari suatu kumpulan), *generalization* (satu *class* merupakan *superclass* dari *class* lainnya), atau *package* (group bersama sebagai satu unit), sebuah sistem biasanya mempunyai beberapa Class Diagram. (UKS, 2017)