

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Telah ada beberapa penelitian yang dilakukan terkait dengan aplikasi kursus pembelajaran bahasa pemrograman dengan android, diantaranya adalah :

Tabel II.1 Penelitian Terkait

No.	Nama	Judul	Hasil Penelitian	Perbedaanya
1.	Erma Susanti, 2015	Rancang Bangun Aplikasi E-Learning	Sistem E-Learning merupakan aplikasi yang diharapkan mampu mendukung proses kegiatan belajar-mengajar. Sehingga melalui E-Learning, proses belajar-mengajar dapat dilakukan dengan cepat dan mudah serta dapat meningkatkan efisiensi dan efektifitas waktu dan biaya	Sistem E-Learning ini masih bersifat sederhana dan belum begitu kompleks sehingga ada beberapa fasilitas yang dinilai masih kurang
2.	Ferry Yudhitama Putra,	Aplikasi Reservasi Les Private Di Easyspeak	Untuk mempermudah pengguna dalam hal reservasi,	Aplikasi ini dirancang dalam dua aplikasi yang berbeda yaitu

	(2016)	Denpasar Berbasis Web Dan Android	maka sistem akan dibangun berbasis web dan Android. Pembangunan aplikasi reservasi les private dibangun dengan bahasa pemrograman PHP dan Java dengan menggunakan framework Codeigniter pada sisi web	berbasis web dan Android yang disertai dengan database untuk menyimpan data, sehingga student dapat melihat informasi tentang les private
--	--------	-----------------------------------	---	---

II.2. Landasan Teori

II.2.1. Pengertian Aplikasi

Program aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Aplikasi juga diartikan sebagai penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan atau sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu. Aplikasi software yang dirancang untuk penggunaan praktisi khusus, klasifikasi luas ini dapat dibagi menjadi 2 (dua) yaitu:

- a. Aplikasi *software* spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
- b. Aplikasi paket, suatu program dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu (Rahmatillah ; 2011: 3).

II.2.1.1. Perkembangan Teknologi Komunikasi dan Informasi

Perkembangan teknologi informasi dan komunikasi (TIK) terakhir mengalami kemajuan yang sangat pesat. Hal tersebut dikarena oleh kuatnya era globalisasi, di mana komputer dan internet dengan sifatnya yang dinamis merupakan fasilitas yang telah mendominasi berbagai aktivitas kehidupan, sehingga aktivitas pendidikan, perkantoran, komersial dan industri secara mutlak memerlukan ketersediaan fasilitas tersebut. Karakteristik masyarakat itu dikenal dengan istilah masyarakat berbasis pengetahuan, artinya komunitas yang menguasai pengetahuan, berpeluang untuk memanfaatkan informasi serta menjadikan informasi sebagai nilai tambah dalam kehidupannya (Syaad Patmanthara ; 2012).

II.2.2. E-Learning (Kursus)

Di dunia pendidikan dan pelatihan sekarang, banyak sekali praktik yang disebut E-Learning. Sampai saat ini pemakaian kata E-Learning sering digunakan untuk menyatakan semua kegiatan pendidikan yang menggunakan media komputer dan Internet. Banyak pula terminologi lain yang mempunyai arti hampir sama dengan E-Learning, diantaranya : Web-based training, online learning, computer-based training/ learning, distance learning, computer-aided instruction, dan lainnya. Terminologi E-Learning sendiri dapat mengacu pada semua kegiatan pelatihan yang menggunakan media elektronik atau teknologi informasi (Erma Susanti; 2015).

II.2.3. Bahasa Pemrograman

Bahasa yang dipakai untuk menginstruksikan computer disebut bahasa pemrograman. Ada 2 jenis bahasa pemrograman terdiri dari bahasa tingkat tinggi & bahasa tingkat rendah. Kita kenal diantaranya: Basic, Algol, Cobol, Pascal, PL-1, RPG, SNOBOL, APL, LISP, GPSS, ADA, DEAL dan sebagainya yang merupakan bahasa tingkat tinggi.

Bahasa yang dimengerti oleh mesin computer adalah intruksi dalam bahasa mesin (Machine Language) yang merupakan bahasa tingkat rendah, jadi bahasa tingkat tinggi yang kita sebutkan diatas agar dapat dimengerti oleh computer haruslah diterjemahkan lebih dahulu oleh kompilator. Ciri dari bahasa tingkat rendah adalah bahwa cara penulisannya intruksinya sangat mendekati bentuk intruksi-intruksi dalam bahasa mesin. Bahasa tersebut dalam bentuk kode HEXA yang sulit bagi sebagian untuk mengertinya.

Komputer bekerja secara elektronik, maka ia tidak dapat menerima masukan berupa tulisan kertas begitu saja. Ia baru dapat membaca informasi dalam bentuk kombinasi bit-bit listrik. Kombinasi-kombinasi tersebut didalam computer sering disebut sebagai kode mesin (Machine kode) yang tidak lain adalah bahasa computer itu sendiri, yaitu kode-kode yang dikenal oleh computer, manusia pun dapat mengerti kodekode tersebut hanya tersusun dari angka nol dan satu yang berderet panjang sekali. Jadi computer sudah mengenal istilah bahasa, yaitu serangkaian kombinasi kode yang digunakan untuk menyampaikan rumus. Didalam bahasa program, perintahperintah/rumus-rumus kebanyakan dituliskan dalam bahasa inggris. Dari bahasa inggris yang sama itu telah dibuat orang

berbagai macam cara memberikan perintah pada computer. (Jusuf Wahyudi : 2013).

II.2.4. Android

Android merupakan sistem operasi berbasis *Linux* yang bersifat terbuka (*open source*) dan dirancang untuk perangkat selular layer sentuh seperti smartphone dan komputer tablet. Android dikembangkan oleh Android, inc. dengan dukungan financial dari google yang kemudian dibeli pada tahun 2005. (Sherief Saibino, 2014 : 7).

1. Sejarah *Android*

Pada bulan oktober 2003 Android, inc. didirikan di Palo Alto, California, oleh Andy Rubin (pendiri Danger), Rich Miner (pendiri Wildfire Communications, inc), Nick Sears (mantan Vp T-Mobile), dan chris White (kepala desain dan pengembangan antar muka web IV) untuk mengembangkan Android yaitu untuk mengembangkan sebuah sistem operasi canggih yang ditujukan untuk kamera digital (Sherief Saibino, 2014 : 8).

2. Fundamental Android

Aplikasi *Android* ditulis dalam bahasa pemrograman *java*. Kode *java* dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, dimana prosesnya di *package* oleh *tools* yang dinamakan “*apt tools*” ke dalam paket *Android* sehingga menghasilkan file dengan ekstensi *apk*. Empat jenis komponen aplikasi *Android*, yaitu :

1) *Activities*

Suatu *activity* akan menyajikan user *interface* (*UI*) kepada pengguna, sehingga pengguna dapat melakukan interaksi.

2) *Service*

Service tidak memiliki *Graphic User Interface* (*GUI*), tetapi *service* berjalan secara background.

3) *Broadcast receiver*

Broadcast receiver berfungsi untuk menerima dan bereaksi untuk menyampaikan notifikasi.

4) *Content Provider*

Content Provider membuat kumpulan aplikasi data secara spesifik sehingga bias digunakan oleh aplikasi lain.

3. Versi Android

Awal sistem android yang dirilis yaitu Android beta bulan November 2007. Sedangkan versi komersial pertama, android 1,0 dirilis pada September 2008. Sejak april 2009, versi Android yang dikembangkan diberi kode nama yang berdasarkan makanan pencuci mulut dan manis. (Sherief Saibino, 2014 : 15).

Tabel. II.1 Versi Android

Versi	Nama	Rillis	Catatan
1.0	<i>Android 1.0</i>	23 September 2008	Android pertama hanya untuk <i>smartphone</i>
1.1	<i>Android 1.1</i>	9 Februari 2008	
1.5	<i>Cupcake</i>	30 April 2009	Mulai pakai kode nama

1.6	<i>Donut</i>	15 September 2009	
2.0- 2.1	<i>Éclair</i>	26 Oktober 2009 (2.0) 12 Januari 2010 (2.1)	
2.2	<i>Froyo</i>	20 Mei 2010	
2.3	<i>Gingerbread</i>	6 Desember 2010	Digunakan pada <i>smartphone</i> jenis lama
3.0- 3.2	<i>Honeycomb</i>	22 Februari 2011 (3.0) 10 Mei 2011 (3.1) 15 Juli 2011 (3.2)	Hanya untuk tablet
4.0	<i>ICS (Ice Cream Sandwich)</i>	19 Oktober 2011	<i>Smartphone</i> dan tablet
4.1- 4.3	<i>Jelly Bean</i>	9 Juli 2012 (4.1) 13 November 2012 (4.2) 24 Juli 2013 (4.3)	<i>Update</i> untuk memperbaiki dan menambah <i>fitur</i> pada <i>ICS</i>
4.4	<i>Kit kat</i>	3 September 2013	
5.0	<i>Lollipop</i>	12 november 2014 (5.0) 9 Maret 2015 (5.1)	
6.0	<i>Marshmallow</i>	5 oktober 015	Terdapat <i>daze mode</i> , <i>Do Not Disturb mode</i> , mendukung <i>USB tipe C</i> , mendukung pembacaan <i>fingerprint</i> .

Sumber: (Sherief Saibino, 2014 : 15)

II.2.5. *Android Studio*

Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Konsep yang dimiliki Android Inc ternyata menggugah minat Google untuk memilikinya. Pada bulan Agustus 2005, Akhirnya Android Inc diakuisisi oleh Google Inc. seluruh sahamnya dibeli oleh Google. Banyak yang memperkirakan nilai pembelian Android Inc Oleh Google adalah sebesar USD 50 juta. saat itu banyak yang berspekulasi bahwa akuisisi ini adalah langkah awal yang dilakukan Google untuk masuk kepasar mobile phone.

Maka dari situlah banyak pengembang sistem maupun software berlomba lomba untuk membuat atau merancang sistem Android menggunakan software – software yang support dengan Android, dan sebagai contoh disini kita pengenalan Android Studio. 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 (google developer ; 2016).

II.2.6. *UML (Unified Modelling Language)*

UML merupakan kependekan dari *Unified Modeling Language* yaitu diagram dan metode standar untuk memodelkan dan merepresentasikan *object oriented software* dan sistem bisnis. (Mulyani, 2016 : 243).

Beberapa fungsi dan kegunaan dari *UML* yaitu (Mulyani, 2016 : 244) :

1. *Visualizing*

Visualizing adalah sebagai alat komunikasi konseptual model antara tim pengembang sistem (sistem analis dengan programmer)

2. *Specifying*

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

3. *Constructing*

Constructing adalah *UML* sebagai bahasa grafis mampu melakukan *mapping* dan konseptual model kedalam bahasa pemrograman.

4. *Documenting*

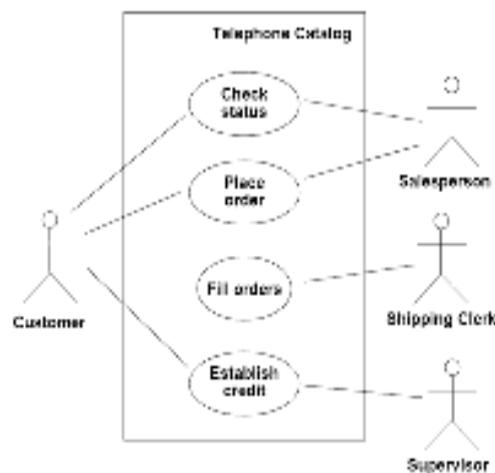
Documenting adalah *UML* digunakan sebagai *tools* untuk melakukan dokumentasi teknis sebuah sistem.

Diagram-diagram yang terdapat dalam *UML* sangat banyak, berikut ini beberapa diagram yang sering digunakan dalam pengembangan sistem yaitu :

1. *Use Case Model*

Use case model merupakan kumpulan diagram dan text yang saling bekerja sama untuk mendokumentasikan bagaimana *user* (aktor) berinteraksi dengan sistem. *Use case model* terdiri dari beberapa diagram :

a) *Use case diagram* yaitu diagram yang menggambarkan dan merepresentasikan aktor, *use cases*, dan *dependencies* suatu proyek dimana tujuan dan diagram ini adalah untuk menjelaskan konsep hubungan antara sistem dengan dunia luar. *Use case diagram* dapat dilihat pada gambar II.1 seperti berikut.



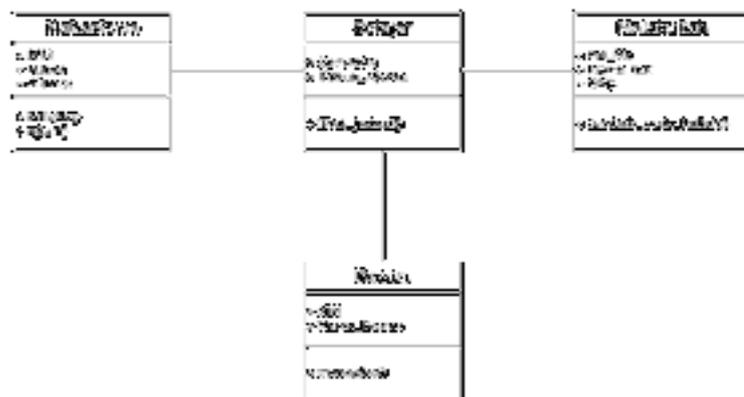
Gambar II.1 Use Case Diagram
(Sumber : Mulyani ; 2016)

b) *Use case narrative* yaitu deskripsi yang menjelaskan use case diagram. Pada *use case diagram* sistem hanya digambarkan secara sederhana menggunakan simbol *use case* yang berhubungan (*relationship*) dengan aktor, sehingga terkadang diperlukan deskripsi yang menjelaskan dan proses tersebut.

c) *Use case scenario* yaitu pemecahan kemungkinan logika pada *use case diagram*.

2. *Class Diagram*

Class diagram adalah diagram yang digunakan untuk merepresentasikan kelas, komponen-komponen kelas dan hubungan antara masing-masing kelas serta mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan *property* dan operasi sebuah kelas serta batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut. *UML* menggunakan istilah fitur sebagai istilah umum yang meliputi *property* dan operasi sebuah kelas. *Class diagram* dapat dilihat pada gambar II.2 seperti berikut.



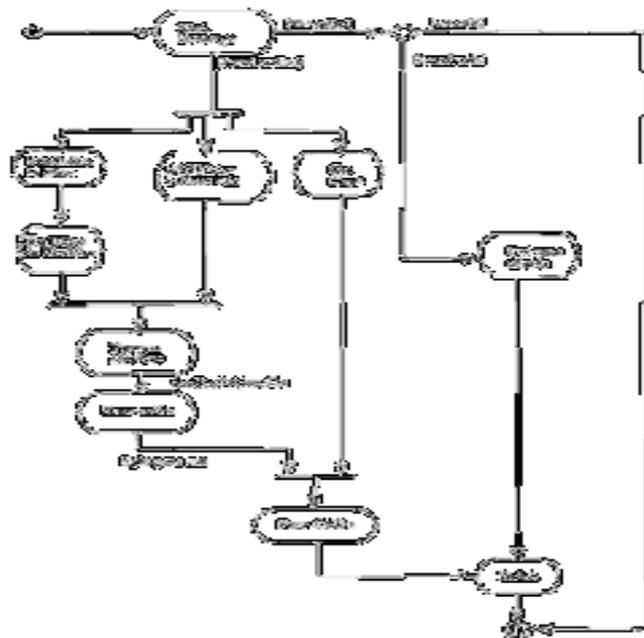
Gambar II.2 *Class Diagram*
(Sumber : Mulyani ; 2016)

3. *Object Diagram*

Object diagram adalah diagram yang digunakan untuk menggambarkan dan merepresentasikan objek dan hubungan antar objek tersebut, selain itu *object diagram* juga dapat digunakan sebagai diagram untuk menunjukkan sebuah konfigurasi contoh dan sebuah objek.

4. *Activity Diagram*

Activity diagram adalah diagram *UML* yang digunakan untuk menggambarkan alur aktivitas dari satu proses. *Activity diagram* memungkinkan siapapun yang melakukan proses untuk memilih urutan dalam melakukannya. Hal ini penting untuk pemodelan bisnis karena proses-proses sering muncul secara paralel. Ini juga berguna pada algoritma yang bersamaan, dimana urutan-urutan independen dapat melakukan hal-hal secara paralel. *Activity diagram* dapat dilihat pada gambar II.3 seperti berikut.



Gambar II.3 Activity Diagram
(Sumber : Mulyani ; 2016)

5. Sequence Diagram

Sequence diagram adalah diagram yang menggambarkan interaksi antar objek. *Sequence diagram* secara khusus menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek ini dalam sebuah *use case*.

II.2.7 Basis Data (Database)

Database adalah kumpulan dari semua data yang diperlukan oleh sistem. Dengan menggunakan database, beberapa aplikasi berbeda bisa saling terintegrasi, misalnya aplikasi keuangan, aplikasi kepegawaian dengan penggajian atau aplikasi persediaan. (Mulyani, 2016 : 148).

Suatu bangunan basis data memiliki jenjang sebagai berikut:

1. Karakter, merupakan bagian data terkecil yang berupa angka, huruf, atau karakter khusus yang membentuk sebuah *item* data atau *field*. Contoh A,B,X,Y,2,1,2,9,0,=,<,> dan sebagainya.
2. *Field/item*, merupakan representasi suatu atribut dan *record* (rekaman/tupel) yang sejenis yang menunjukkan suatu *item* dari data. Contoh *field* nama (berisi data nama-nama pegawai) dan lain sebagainya.
3. *Record/rekaman/tupel*: Kumpulan dari field membentuk suatu record atau rekaman. Record menggambarkan suatu unit data individu yang tertentu. Contoh: file pegawai, dimana tiap-tiap *record* berisi kumpulan data nama, alamat, departemen, yang dapat mewakili tiap-tiap data.
4. *File*, merupakan kumpulan dari *record-record* yang menggambarkan satu kesatuan data yang sejenis. Contoh *file* pegawai berisi data tentang semua yang berhubungan dengan pegawai seperti nama pegawai, alamat pegawai, departemen, yang dapat mewakili tiap-tiap data.
5. *Database*, merupakan kumpulan dari *file* atau *tabel* yang membentuk suatu *database*. Contoh *database* pegawai berisi file pegawai dan sebagainya.