

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Marjito, dkk (2016), mengenai aplikasi penjualan online berbasis android. Marjito menyimpulkan bahwa dengan dibuatnya aplikasi ini untuk Toko Hoax Merch, Toko tersebut dapat lebih mudah dalam memasarkan dan menjual produk serta dapat membantu memberikan informasi seputar produk kepada pembeli. Menurut Marjito, dengan dibuatnya aplikasi penjualan online pada Toko Hoax Merch juga dapat mempermudah konsumen untuk melihat produk apa saja yang tersedia di Toko Hoax Merch.

Berdasarkan penelitian yang dilakukan oleh Nurbo Jatmiko, dkk (2010), mengenai analisa dan perancangan sistem informasi dalam layanan penjualan berbasis *web* pada Mall Puri Indah. Nurbo Jatmiko menyimpulkan bahwa dengan dibuatnya *Website* Mall Puri Indah ini dapat mempermudah pihak Mall Puri Indah dalam meng-update informasi yang akan disampaikan, serta menjadi sarana untuk mempromosikan Mall Puri Indah dengan memberikan daya tarik tersendiri karena didukung oleh unsur – unsur multimedia seperti animasi, suara, teks, gambar, dan lain – lain sehingga dapat menarik lebih banyak pengunjung.

Berdasarkan penelitian yang dilakukan oleh Arif Rizki, dkk (2017), mengenai Implementasi Aplikasi Layanan Informasi Lokasi Strategis Kota Payakumbuh Berbasis Android. Arif menyimpulkan bahwa dengan dibuatnya aplikasi ini, akan

membantu dan mempermudah para pengunjung untuk mengetahui informasi lokasi-lokasi strategis kota Payakumbuh yang mau mereka tuju.

Berdasarkan penelitian yang dilakukan oleh M Affan Effendi, dkk (2011), mengenai Perancangan Sistem Layanan Informasi Akademik Berbasis Short Message Service. M Affan menyimpulkan bahwa aplikasi ini memberikan kemudahan untuk mendapatkan dengan cepat informasi-informasi pokok yang diperlukan dalam kaitannya dengan proses belajar mengajar yang dijalaninya.

Berdasarkan penelitian yang dilakukan oleh Debasatwa Dutta (2017), mengenai Developing an Intelligent Chat-bot Tool to assist high school students for learning general knowledge subjects. Debasatwa membandingkan berbagai macam *platform chatbot* mulai dari API.AI, Wit.Ai, Luis.Ai, dan Pandorabots. Dan dari hasil yang dibandingkan tersebut, Debasatwa memilih API.AI sebagai *platform* untuk membangun *chatbot* cerdas dalam membantu siswa sekolah menengah untuk mempelajari pengetahuan umum atau mata pelajaran ilmu sosial mereka.

Perbedaan penelitian diatas dengan penelitian yang penulis lakukan terletak pada studi kasus yang penulis ambil yaitu Aplikasi Layanan Informasi Penjualan pada PT. Ouzen Anugerah Indonesia. Selain itu juga terdapat perbedaan pada *platform* yang penulis gunakan yaitu API.AI yang merupakan *platform chatbot* yang menyediakan layanan NLP (*Natural Language Processing*) dan NLU (*Natural Language Understanding*), serta aplikasi ini berbasis Android.

II.2. Landasan Teori

II.2.1. Aplikasi

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju (Andi Juansyah, 2015).

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan (Hasan & Asep, 2014).

Dengan begitu dapat diambil kesimpulan bahwa aplikasi adalah sebuah program yang sudah siap digunakan dan memudahkan pengguna dalam melakukan aktivitas yang difokuskan kepada tujuan aplikasi tersebut dibuat.

II.2.2. *Layanan Informasi*

Layanan informasi adalah layanan bimbingan konseling yang memungkinkan peserta didik (klien) menerima dan memahami berbagai informasi (seperti informasi pendidikan dan informasi jabatan) yang dapat digunakan sebagai bahan pertimbangan dan pengambilan keputusan untuk kepentingan peserta didik (klien) (Ledy & M. Arli, 2016).

Layanan informasi adalah kegiatan bimbingan yang bermaksud membantu siswa untuk mengenal lingkungannya, yang sekiranya dapat dimanfaatkan untuk masa kini maupun masa yang akan datang (Edy Sutrisno, 2017).

Layanan informasi adalah kegiatan bimbingan yang bermaksud membantu siswa untuk mengenal lingkungannya. Tujuan layanan informasi ini untuk membekali individu dengan berbagai hal yang berguna untuk mengenal diri, merencanakan dan mengembangkan pola kehidupan sebagai pelajar, anggota keluarga dan masyarakat. Pemahaman yang diperoleh melalui layanan informasi, digunakan sebagai bahan acuan dalam meningkatkan kegiatan dan prestasi belajar, mengembangkan cita-cita, menyelenggarakan kehidupan sehari-hari dan mengambil keputusan (Listianah, 2013).

Dengan begitu dapat diambil kesimpulan bahwa layanan informasi adalah memberikan arahan dan bimbingan untuk memudahkan pengguna dalam mengetahui informasi mengenai produk atau layanan yang ditawarkan.

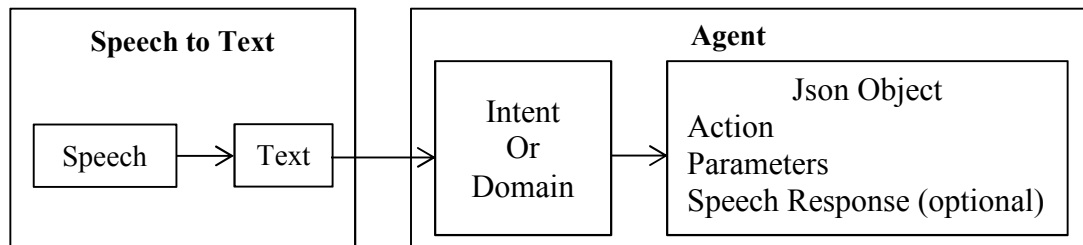
II.2.3. *Penjualan*

Sedangkan penjualan adalah suatu usaha yang terpadu untuk mengembangkan rencana-rencana strategis yang diarahkan pada usaha pemuasan kebutuhan dan keinginan pembeli, guna mendapatkan penjualan yang menghasilkan laba. Penjualan adalah penerimaan yang diperoleh dari pengiriman barang dagangan atau dari penyerahan pelayanan dalam bursa sebagai barang pertimbangan. Pertimbangan ini dalam bentuk tunai peralatan kas atau harta lainnya. Pendapatan dapat diperoleh pada

saat penjualan, karena terjadi pertukaran, harga jual dapat ditetapkan dan bebannya diketahui (Muhammad Wyzer et al, 2011).

II.2.4. *API.AI*

Api.ai yang sebelumnya dikenal sebagai “SpeakToIt” adalah sebuah layanan *Natural Language Processing* yang disediakan oleh Google Inc. Api.ai mendukung *Speech to text* dan juga kemampuan *Text to Speech*.



Gambar II.1 Struktur API.AI

Api.ai memiliki dukungan untuk *Intents*, *Entities*, *Action* bersama dengan area focus untuk *Domain*. Api.ai memiliki daftar *Entities* yang sudah ada sebelumnya seperti @sys.data, @sys.color, @sys.currency, @sys.units, dan sebagainya. *Entities* ini dapat mengartikan perilaku manusia dan merespons kembali ke pengguna dengan bahasa yang interaktif. Saat ini, *Agent* yang sudah ada sebelumnya tersedia dalam bahasa Inggris, Spanyol, Prancis, Jepang, Italia, Portugis, Portugis Brasil, Jerman, Rusia, Cina (Sederhana), Cina (Kanton), dan Cina (Tradisional).

Api.ai juga menyediakan modul integrasi untuk aplikasi pihak ketiga langsung ke *Facebook messenger*, *Line*, *Skype*, dll. Membangun aplikasi Api.ai bisa digunakan pada layanan berbayar perusahaan Google dan juga dapat digunakan secara gratis di aplikasi Heroku atau *server* serupa lainnya. (Kunal Shah & Flavia Gonsalves, 2017).

Berikut ini penulis akan menjelaskan lebih lanjut lagi mengenai istilah yang penulis gunakan dalam membuat aplikasi layanan informasi penjualan pada PT. Ouzen Anugerah Indonesia menggunakan API.AI berbasis android.

a. *Agent*

Agent adalah sebutan untuk sebuah yang akan dibuat pada API.AI untuk aplikasi yang ingin penulis buat.

b. *Intent*

Intent adalah sebutan untuk pemetaan dari percakapan antara pengguna dalam hal ini adalah pelanggan dengan aplikasi. *Intent* ini berfungsi untuk membuat aplikasi paham akan pertanyaan atau kalimat yang di-*input* oleh pengguna. Pada setiap *Intent* juga terdapat pilihan untuk menambahkan *slot filling*, yaitu kondisi dimana aplikasi akan meminta pengguna untuk mengisi *Entities* yang dimaksud.

c. *Entities*

Entities adalah sebutan untuk objek dalam *input* yang diberikan oleh pengguna kepada aplikasi yang bertujuan untuk lebih memahami apa maksud permintaan dari pengguna. Salah satu contoh seperti yang akan penulis jelaskan pada Bab 3 yaitu mengenai *entities* “produk” yang digunakan untuk menyimpan kata kunci berupa daftar produk yang dimiliki oleh OuzenMart.

II.2.5. *Android*

Android adalah suatu sistem operasi yang berbasis Linux dengan perangkat *mobile device*, android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi-aplikasi baru untuk membantu kebutuhan manusia. Awal terdirinya android dimulai dari Google Inc yang membeli Android inc pendatang bat dari perangkat lunak untuk ponsel/smartphone. Semakin banyak smartphone dan PC Tablet menggunakan sistem operasi dengan versi yang berbedda. Semakin tinggi versinya maka fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi Android adalah HTC Dream yang dirilis pada tanggal 22 Oktober 2008. Perkembangan sistem android berkembang sesuai versi keluarnya adapun versi android dimulai dari (Muhamad Zaelani ; 2017: 5) :

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> .
7.0	<i>Nougat</i>	22 Agustus 2016	Multitasking

Tabel. II.1 Versi Android

II.2.5. Java

Java adalah sebuah bahasa pemrograman yang populer dikalangan para akademisi dan praktisi komputer. Java pertama kali dikembangkan untuk memenuhi kebutuhan akan sebuah bahasa komputer yang ditulis satu kali dan dapat dijalankan dibanyak system komputer berbeda tanpa perubahan kode berarti. Pada umumnya, para pakar pemrograman berpendapat bahwa bahasa Java memiliki konsep yang konsisten dengan teori pemrograman objek dan aman untuk digunakan. Java sampai saat ini masih merupakan bahasa pemrograman yang masih sangat diminati dan banyak digunakan oleh para progremer dan software developer untuk mengembangkan berbagai tipe aplikasi, mulai dari aplikasi *console*, aplikasi *desktop*, *game*, dan *applet* (aplikasi yang berjalan di lingkungan *web browser*), sampai ke aplikasi aplikasi yang berskala *enterprise*. Untuk memenuhi kebutuhan tipe aplikasi

yang beragam tersebut, Java dikategorikan menjadi tiga edisi, yaitu: J2SE (*Java 2 Platform Standard Edition*) untuk membuat aplikasi-aplikasi desktop dan applet, J2EE (*Java 2 Platform Enterprise Edition*) untuk membuat aplikasi-aplikasi *multitier* berskala *enterprise*, dan J2ME (*Java 2 Platform Micro Edition*) untuk membuat aplikasi-aplikasi yang dapat dijalankan di lingkungan perangkat-perangkat mikro seperti handphone, PDA dan Smartphone. (Retno & Husnul, 2013).

II.2.6. Firebase

Firestore adalah sebuah penyedia layanan berupa database *realtime* dan *backend* yang dapat digunakan pada berbagai platform. *Backend* sendiri adalah sebuah bagian dalam code aplikasi yang berhubungan langsung dengan isi *database*. Dengan Firestore, pengembang aplikasi tidak perlu membuat *backend* sendiri melainkan memakai API yang telah disediakan oleh Firestore sehingga pengembangan aplikasi dapat dipersingkat. Firestore dikembangkan dengan menggunakan database MongoDB sehingga Firestore menggunakan tipe database NoSQL. Karena memakai tipe database NoSQL maka struktur database dari Firestore bersifat fleksibel dan cepat sehingga cocok untuk digunakan pada aplikasi berbasis *mobile* (Sontama et al, 2015).

Firestore adalah platform seluler yang membantu developer mengembangkan aplikasi berkualitas tinggi secara cepat, berbasis pengguna, dan dapat menghasilkan uang lebih banyak. Firestore terdiri dari fitur pelengkap yang bisa dipadupadankan sesuai dengan kebutuhan Anda (Google). Digunakannya firestore, karena platform yang telah dikembangkan oleh Google ini memiliki fitur cukup mumpuni untuk

dimanfaatkan pada aplikasi messenger. Dalam mengembangkan aplikasi brawijaya messenger ini fitur yang mungkin digunakan dari firebase adalah FCM (*Firebase Cloud Messaging*), *Authentication*, *Realtime Database*, dan *Firestore Storage*. FCM merupakan pengembangan dari GCM (*Google Cloud Messaging*) yang dapat dimanfaatkan untuk multiplatform, beberapa keunggulan dari FCM adalah *upstream/downstream* tanpa batas, mengatasi aspek antrian dan pengiriman. Sedangkan *Authentication* dapat membantu meningkatkan tingkat keamanan permasalahan verifikasi dan validasi akun, ditambah lagi fitur ini dapat memungkinkan developer untuk mengintegrasikan media terkenal seperti *Facebook*, *Google*, *Twitter* dengan sistem yang dikembangkan. Terakhir *Realtime Database* dan *Storage* adalah fitur yang sangat membantu untuk membuat suatu aplikasi yang bersifat dinamis, karena dengan memanfaatkan fitur ini suatu aplikasi dapat secara dinamis menerima masukan dan membagikannya dengan *realtime* serta menyimpannya kedalam *storage* secara bersamaan (Aditya et al, 2018).

II.2.7. UML(Unified Modeling Language)

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

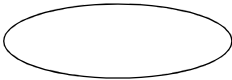
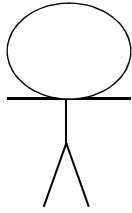
Menurut Gellysa Urva dan Helmi Fauzi Siregar (2015) UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan



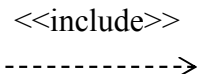
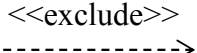
alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.2 dibawah ini:



Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, dan dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak</p>


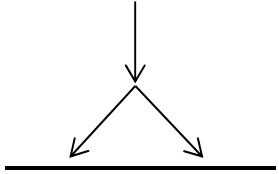
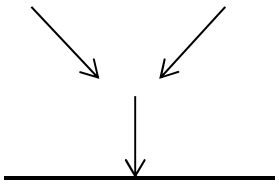
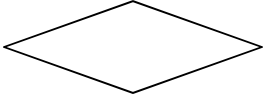

	memiliki <i>control</i> terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidिकासikan aliran data.
	Generalisasi menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan <i>use case</i>
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan tambahan fungsional dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

Tabel II.2. Simbol Use Case
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015 : 94)

2. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.3 dibawah ini:

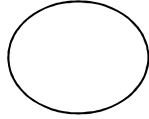
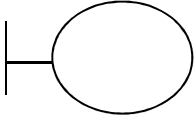
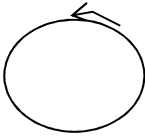

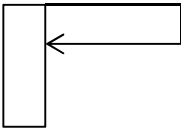

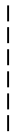
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.

	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , untuk menunjukkan siapa melakukan apa.

Tabel II.3. Simbol *Activity Diagram*
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015 : 94)

3. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.4 dibawah ini :

Gambar	Keterangan
	<i>EntityClass</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

Tabel II.4. Simbol *Sequence Diagram*
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015: 95)

4. *Class Diagram*

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.5 dibawah ini:

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

Tabel II.5. *Multiplicity Class Diagram*
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar; 2015: 95)