

## **BAB 2**

### **TINJAUAN PUSTAKA**

Pada tinjauan pustaka, penulis akan menerangkan teori-teori umum informatika yang berhubungan dengan tema penelitian ini.

#### **2.1. Profil Instansi**

Profil instansi akan memberikan gambaran tentang sejarah instansi, logo STIKES Bhakti Kencana Bandung, tujuan instansi, visi dan misi sekolah dan struktur organisasi dari tempat penelitian ini.

##### **2.1.1. Sejarah Instansi**

Sekolah Tinggi Ilmu Kesehatan Bhakti Kencana Bandung berdiri tanggal 02 Januari 2002 dengan SK Dirjen Dikti No. 1/D/0/2002 dan rekomendasi Depkes RI No. KS.02.1.5.4327 dibawah naungan Yayasan Adhi Guna Kencana Bandung sesuai dengan akta notaris Tien Norman Lubis, S.H. No. 19 tanggal 28 Desember 1998. kemudian diperbaharui melalui SK Notaris Dendi Sefandi, S.H., M.Kn. Nomor 4 tahun 2007 dan mendapat pengesahan dari Menteri Hukum dan HAM nomor: AHU-1406.AH.01.04 Tahun 2009 Membuka 3 program studi, yaitu: Kebidanan, Keperawatan, dan Kesehatan Masyarakat.

#### **Rencana Jangka Pendek:**

- a. Menetapkan struktur organisasi lengkap dengan uraian tugasnya yang mampu menampung pelaksanaan fungsi – fungsi administrasi, hubungan dengan alumni dan fungsi yang mencerminkan ciri khas STIKes Bhakti Kencana Bandung.
- b. Menjalin hubungan dengan lembaga – lembaga pendidikan tinggi di dalam dan di luar negeri.
- c. Membenahi dan meningkatkan aktivitas bidang penelitian.
- d. Meningkatkan keterampilan tenaga penelitian.
- e. Pengadaan / peningkatan jumlah tenaga pengelola penelitian.
- f. Membentuk / memantapkan organisasi pengabdian pada masyarakat.

- g. Pengadaan sarana yang memadai seperti kelengkapan buku perpustakaan, peralatan kantor, ruang dan sebagainya dalam rangka meningkatkan kelancaran pelaksanaan bidang pengabdian pada masyarakat.
- h. Meningkatkan kesadaran civitas akademik untuk lebih berperan aktif di bidang pengabdian pada masyarakat dan kemampuan mengelola aktivitas yang terkait bidang pengabdian pada masyarakat.
- i. Melaksanakan pendekatan kepada masyarakat dan komunikasi dengan berbagi instansi dalam rangka pelaksanaan magang kerja mahasiswa.

#### **Rencana Jangka Panjang:**

- a. Pembinaan staf peneliti.
- b. Peningkatan pengadaan dana peneliti baik yang bersumber dari akademik atau sumber lain.
- c. Menyebarkan hasil penelitian dengan mengadakan diskusi dan seminar atau melalui jurnal nasional maupun media lainnya.
- d. Menindaklanjuti program penelitian tahap pertama yang belum terselesaikan.
- e. Berperan serta secara aktif dalam proses pembangunan dalam rangka mengamalkan ilmu pengetahuan, teknologi dan seni yang sudah dikuasai.
- f. Pengembangan kegiatan kemahasiswaan.
- g. Pengembangan kerumahtanggaan

#### **2.1.2. Tujuan Instansi**

Berikut ini adalah tujuan dari STIKES Bhakti Kencana Bandung:

1. Menyelenggarakan proses pembelajaran untuk mempersiapkan peserta didik menjadi anggota masyarakat yang mempunyai kemampuan bidang akademik dan profesional serta dapat menciptakan atau menerapkan ilmu pengetahuan dan teknologi di bidang kesehatan.
2. Mengembangkan dan menyebarkan ilmu pengetahuan dan teknologi bidang kesehatan dalam upaya meningkatkan taraf hidup masyarakat sesuai dengan nilai-nilai agamis.

3. Menyelenggarakan proses pembelajaran untuk mempersiapkan peserta didik menjadi masyarakat profesi yang memiliki kemampuan akademik dan profesional dalam menunjang praktik mandiri yang agamis.
4. Pengembangan staf akademik dengan kompetensi ke arah yang professional

Selanjutnya, berikut tujuan dari program studi kebidanan:

1. Terwujudnya ahli madya kebidanan yang beriman, bertaqwa, dan berakhlak mulia.
2. Terciptanya lulusan yang memiliki kemampuan akademik, profesional serta berdedikasi tinggi demi terwujudnya masyarakat yang sehat dan berkualitas.
3. Menghasilkan peserta didik yang memiliki keunggulan dalam memberikan pelayanan asuhan kebidanan dengan pendekatan gentle birth dan asuhan kebidanan kegawatdaruratan maternal neonatal sesuai dengan kewenangannya.
4. Mengamalkan, mengembangkan, menciptakan dan menyebarluaskan ilmu pengetahuan, teknologi dalam rangka meningkatkan kesejahteraan masyarakat, bangsa dan negara.
5. Menyelenggarakan Tri Dharma Perguruan Tinggi yang berkualitas.

### **2.1.3. Visi dan Misi Instansi**

Adapun Visi Sekolah adalah sebagai berikut:

Menjadikan STIKES Bhakti Kencana sebagai perguruan tinggi yang unggul, mandiri, dalam penyelenggaraan Tri Dharma Perguruan Tinggi untuk menciptakan sumber daya manusia yang Profesional dan mampu bersaing secara global 2017.

Sedangkan Misi Sekolah adalah sebagai berikut:

1. Menyelenggarakan proses pendidikan yang menghasilkan sumber daya manusia yang handal, profesional dan mandiri sesuai kebutuhan pembangunan.
2. Melaksanakan penelitian bidang kesehatan dan pengabdian kepada masyarakat dalam meningkatkan derajat kesehatan.
3. Melengkapi sarana dan prasarana untuk menunjang proses belajar mengajar.

4. Meningkatkan kualitas sumber daya manusia melalui pengembangan staff akademik sesuai kompetensi dan perkembangan IPTEK.
5. Menjalin kerjasama dengan berbagai pihak dalam rangka pengembangan Tri Dharma Perguruan Tinggi baik di dalam maupun di luar negeri.

**2.1.4. Logo Instansi**

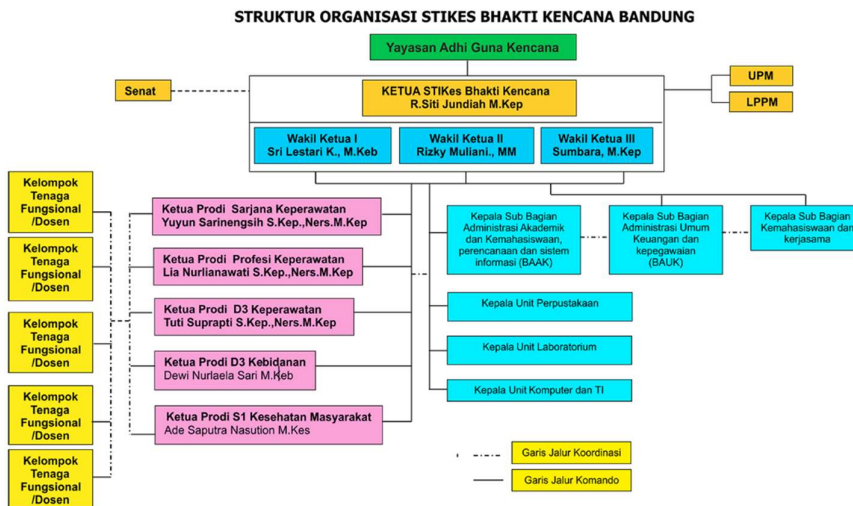
Berikut ini adalah logo STIKES Bhakti Kencana seperti terlihat pada gambar 2.1.



**Gambar 2.1 Logo STIKes Bhakti Kencana Bandung**

**2.1.5. Struktur Organisasi Instansi**

Berikut ini adalah struktur organisasi STIKES Bakti Kencana seperti yang terlihat pada gambar 2.2.



**Gambar 2.2 Struktur Organisasi STIKes Bhakti Kencana Bandung**

## 2.2. Kecerdasan Buatan

Kecerdasan buatan (*Artificial Intelligence*) menurut H. A. Simon [5] merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas. Definisi kecerdasan buatan lainnya merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia.

Kecerdasan buatan dilihat dari berbagai sudut pandang adalah sebagai berikut:

1. Sudut pandang kecerdasan

Kecerdasan buatan yaitu proses bagaimana membuat mesin yang "cerdas" dan dapat melakukan hal-hal yang sebelumnya dapat dilakukan oleh manusia.

2. Sudut pandang penelitian

Studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dilakukan oleh manusia.

3. Sudut pandang bisnis

Kumpulan peralatan yang sangat *powerful* dan metodologis dalam menyelesaikan masalah-masalah bisnis.

4. Sudut pandang pemrograman

Kecerdasan buatan termasuk didalamnya adalah studi tentang pemrograman simbolik, pemecahan masalah, serta proses pencarian.

Ada tiga tujuan kecerdasan buatan, diantaranya: membuat komputer lebih cerdas, mengerti tentang kecerdasan, serta membuat mesin yang lebih berguna. Yang dimaksud kecerdasan adalah kemampuan untuk belajar dan mengerti berdasarkan pengalaman, memahami pesan yang kontradiktif dan ambigu, menanggapi dengan cepat dan baik atas situasi yang baru, menggunakan penalaran dalam memecahkan masalah, serta menyelesaikannya dengan efektif. Adapun subdisiplin ilmu dalam kecerdasan buatan adalah:

1. Sistem Pakar (*Expert System*).  
Komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.
2. Pengolahan Bahasa Alami (*Natural Language Processing*)  
Dengan adanya pengolahan bahasa alami diharapkan user dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
3. Pengenalan Ucapan (*Speech Recognition*)  
Melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan komputer menggunakan suara.
4. Robotika dan Sistem Sensor (*Robotics and Sensory Systems*)  
Robotika adalah suatu bidang rekayasa yang mencurahkan perhatiannya ke bidang duplikasi kemampuan fisik manusia, yaitu suatu komplemen alami dalam bidang artificial intelligence yang berusaha menirukan kemampuan mental manusia. Robot adalah mesin atau manipulator yang mampu melaksanakan fungsi fisik manusia secara terbatas
5. Pengolahan Citra (*Computer Vision*)  
*Computer Vision* mencoba untuk dapat menginterpretasikan gambar atau objek objek tampak melalui komputer.
6. *Intelligent Computer-aided Instruction*  
Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
7. *Game Playing*  
Kebanyakan permainan dilakukan dengan menggunakan sekumpulan aturan (*rule*). Dalam permainan digunakan apa yang disebut dengan pencarian ruang. Teknik untuk menentukan alternatif dalam menyimak problema ruang merupakan sesuatu yang rumit. Teknik tersebut disebut dengan *Heuristic*. Permainan merupakan bidang yang menarik dalam studi *heuristic*.

### 2.2.1. Chatbot

*Chatbot* merupakan program komputer yang dirancang untuk dapat melakukan interaksi dengan manusia melalui pesan teks, maupun suara. *Chatbot* biasanya juga dibekali dengan kecerdasan buatan dan pemrosesan bahasa alami yang membuatnya menjadi program komputer yang cerdas dan dapat menjawab pertanyaan yang diberikan oleh manusia. *Chatbot* dibangun sesuai dengan topik yang sudah dimodelkan dalam basis pengetahuan. Banyak *chatbot* yang sudah ada dibangun sesuai dengan topik dan permasalahan yang ingin dipecahkan oleh seseorang untuk keperluan pribadi ataupun keperluan bisnis, sebagai contoh adalah *chatbot* yang menangani pemesanan pizza. Di dalam *chatbot* tersebut telah ditanamkan model pengetahuan untuk menjawab pertanyaan-pertanyaan yang sesuai dengan konteks yang telah disusun.

Pada mulanya, program komputer (*bots*) ini diuji melalui *Turing Test*, yaitu dengan merahasiakan identitasnya sebagai mesin sehingga dapat mengelabui orang yang bercakap-cakap dengannya. Jika pengguna tidak dapat mengidentifikasi *bots* sebagai suatu program komputer, maka chatterbot tersebut dikategorikan sebagai kecerdasan buatan (*artificial intelligence*). Zaman sekarang ini, *chatterbot* telah dimanfaatkan untuk tujuan praktis seperti bantuan *online*, layanan personal, atau akuisisi informasi, dalam hal ini dapat dilihat fungsi program sebagai suatu jenis agen percakapan (*conversational agent*) [6].

*Chatbot* terdiri dari tiga kombinasi, di mana ketiga kombinasi inilah yang membentuk sebuah chatbot [7], di antaranya adalah:

- a. Tampilan (*User Interface*)

*User interface* dalam *chatbot* ini sendiri adalah jembatan antara *chatbot* dan pengguna saling berinteraksi. Melalui aplikasi pesan berbasis teks. *User Interface* haruslah dapat memberikan pengalaman yang lebih baik kepada pengguna ketika berinteraksi dengan *chatbot*.

b. Kecerdasan Buatan (*Artificial Intelligence*)

Artificial Intelligence atau AI akan membuat *chatbot* mengerti dan memahami setiap interaksi yang terjadi dengan pengguna. *Chatbot* menangani pemecahan masalah melalui aturan yang telah ditentukan sebelumnya di pohon keputusan.

c. Integrasi

Integrasi dengan sistem lainnya akan menambah kekayaan fitur yang terdapat di dalam suatu *chatbot*. Dengan mengintegrasikan *chatbot* ke sistem yang lain dapat menyediakan informasi tambahan. Dengan cara ini *chatbot* mampu memberikan informasi yang lebih kaya kepada user, seperti pada penelitian kali ini yang akan mengimplementasikan *chatbot* pada *order management system*.

*Chatbot* dapat diimplementasikan untuk bidang komersial, pendidikan, hiburan, dan sektor pelayanan publik [8]. *Chatbot* berperan sebagai agen percakapan yang memiliki basis pengetahuan yang dapat digunakan untuk melakukan percakapan dengan penggunanya. Selain itu, *chatbot* juga dapat digunakan sebagai media pembantu pembelajaran bagi siswa sekolah menengah atas pada mata pelajaran pengetahuan umum [9], dimana *chatbot* berperan sebagai asisten siswa untuk perbantuan dalam belajar mata pelajaran umum. *Chatbot* dibangun menggunakan beberapa platform yaitu Dialogflow.com (Api.ai), Wit.ai, Luis.ai, dan Pandorabots.com.

### 2.3. Platform Dialogflow (Api.ai)

Dialogflow atau sebelumnya dikenal Api.ai adalah sebuah *platform* yang menyediakan layanan *Natural Language Processing* (NLP) dan *Natural Language Understanding* (NLU). Layanan ini digunakan untuk membuat *chatbot* lebih cerdas dan dapat memahami maksud dari apa yang ditanyakan oleh pengguna. *Natural language Processing* adalah salah satu disiplin ilmu dari Kecerdasan Buatan (*Artificial Intelligence*) yang fokus terhadap interaksi manusia dan komputer melalui bahasa alami yang biasa manusia gunakan. Dalam NLP tujuan yang ingin dicapai adalah kemampuan sebuah sistem NLP memiliki pengetahuan bahasa alami baik dari susunan kalimat, arti dari kata tersebut dan maksud dari sebuah kalimat.



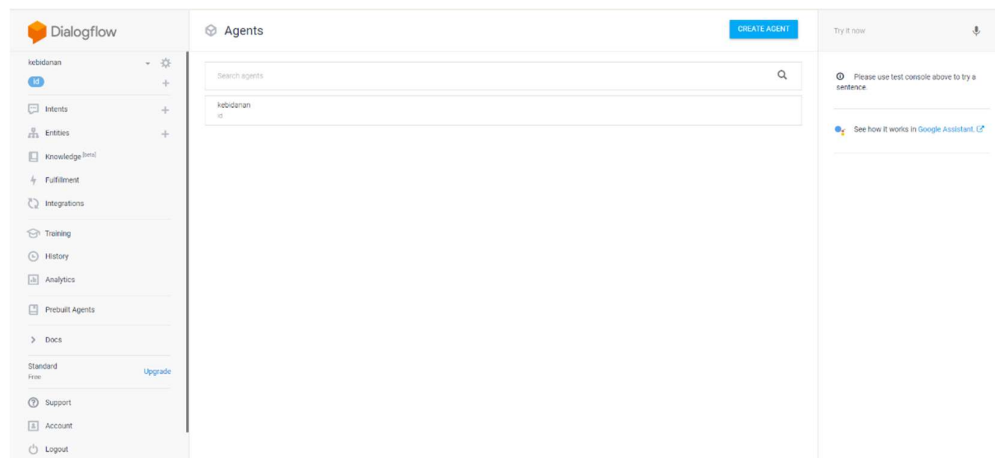
Sedangkan NLU sendiri adalah merupakan sub bidang dari NLP, di mana fokus tujuan dari NLU itu sendiri adalah untuk melakukan pemahaman terhadap suatu kalimat dan melakukan analisis semantik. Dialogflow ini sendiri mendukung sebanyak 20 bahasa, diantaranya adalah Bahasa Indonesia. Namun dialogflow memiliki lebih sedikit dukungan domain berbahasa Indonesia dibandingkan dukungan domain berbahasa Inggris. Domain adalah koleksi pengetahuan dan struktur data. Pada Dialogflow sendiri telah banyak basis pengetahuan yang sudah tertanam di sistem layanan Dialogflow ini. Koleksi pengetahuan yang terdapat pada layanan ini di antaranya adalah entitas musisi, bandara, dll.

### **2.3.1. Sejarah Singkat Dialogflow (Api.ai)**

Dialogflow atau yang dulunya API.AI dan Speaktioit adalah pengembang teknologi interaksi komputer dan manusia. Fokus teknologi yang digunakan adalah dalam bidang *natural language processing* atau NLP. Perusahaan ini telah mengembangkan beberapa produk yang memiliki fitur utama melakukan tugas tanya jawab dengan bahasa alami antara manusia dan komputer. Pada bulan September 2014 Speaktioit memperkenalkan sebuah *platform* yang diberi nama Api.ai. *Platform* inilah yang banyak digunakan oleh pengembang pihak ketiga untuk membuat sebuah *chatbot*. Selain itu Api.ai menyediakan kepada pengembang fitur-fitur lainnya yaitu sebuah *Software Development Kit* (SDK) yang memiliki fungsi dapat menambahkan pengenalan suara, pemahaman bahasa alami dan konversi *text-to-speech* untuk aplikasi Android, dan iOS. Api.ai juga memiliki halaman website yang digunakan untuk melakukan pengujian skenario yang dapat digunakan para developer untuk menguji skenario yang telah dirancang sebelumnya. Pada bulan September tahun 2016 Google membeli perusahaan ini dan mengganti namanya menjadi Dialogflow. Google menggunakan layanan ini untuk asisten virtual Google.

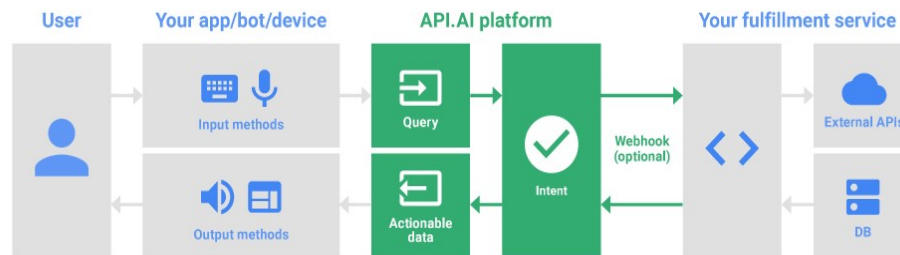
### 2.3.2. Lingkungan Pengembangan

Untuk dapat menggunakan layanan ini, maka penulis perlu melakukan observasi dan belajar untuk mengetahui lingkungan pengembangan dari *platform* Dialogflow. Penulis juga melihat beberapa tutorial untuk dapat membangun sebuah *chatbot* yang terintegrasi dengan *platform* ini. Ada beberapa hal yang perlu penulis ketahui untuk dapat menggunakan layanan ini. Langkah pertama yang penulis lakukan adalah untuk mengetahui bagaimana cara kerja dasar untuk menggunakan layanan ini. Gambar 2.3 menunjukkan halaman *dashboard* dari *console* Dialogflow.



**Gambar 2.3 Tampilan *Dashboard Console* pada *platform* Dialogflow**

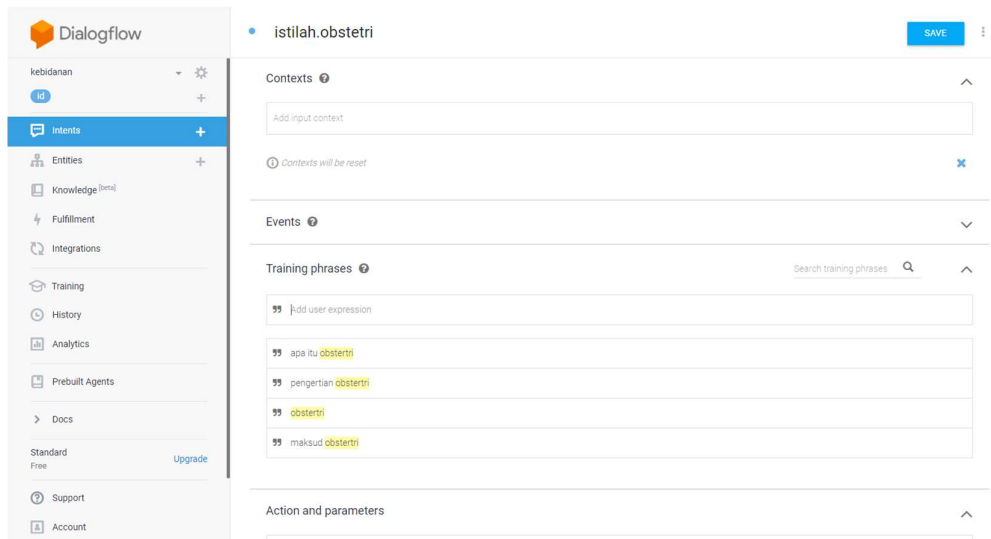
Gambar 2.3 di atas merupakan halaman *dashboard console* pada *platform* Dialogflow. Pada laman *dashboard* ini, *developer* dapat merancang *Conversational Flow* dari permasalahan yang ingin dipecahkan. Proses awal yang dilakukan adalah dengan adanya *request* yang datangnya dari pengguna dengan melakukan permohonan ke *platform*, lalu sistem didalam *platform* ini akan memberikan respon ke pengguna yang terkait bahwa permohonan untuk menggunakan layanan diterima, pengguna di sini dapat dikatakan sebagai *developer* yang merancang sebuah . Gambar 2.4 menunjukkan ilustrasi mengenai alur proses Dialogflow serta relasinya terhadap berbagai komponen lainnya.



**Gambar 2.4 Alur proses Dialogflow serta relasinya terhadap komponen lainnya**

Pertama dimulai dari pengguna yang mengirimkan *request* berupa teks ataupun suara ke dalam *platform* Dialogflow, kemudian *query* atau permintaan tadi akan diproses di dalam *intents* untuk memetakan permintaan tersebut dan tindakan apa yang harus dilakukan. Selanjutnya apabila terdapat informasi tambahan yang akan diberikan sebagai *response*, *platform* ini akan mengirimkan *fulfillment*, ini akan mendapatkan informasi dari sumber daya luar. Lalu pengguna akan mendapatkan jawaban dari pertanyaan tersebut.

Pada langkah awal *developer* akan membuat sebuah *Agent* yang akan digunakan sebagai modul. *End-user* atau pengguna yang akan memanfaatkan layanan *chatbot* yang sudah terintegrasi dengan platform Dialogflow akan menanyakan sesuatu. Permintaan dari *end-user* ini akan diteruskan ke dalam *Agent* yang telah dibuat oleh *developer*. Di dalam *Agent* atau modul inilah permintaan yang datang akan diproses. Agar *Agent* dapat memahami maksud dari permintaan yang dikirim, *Agent* perlu memiliki beberapa sampel atau data latih yang berkaitan dengan pertanyaan tersebut. Oleh karena itu *developer* perlu mendefinisikan terlebih dahulu pertanyaan-pertanyaan yang akan memiliki kesamaan dengan pertanyaan yang dikirim *end-user* ke dalam sebuah *intent*. Semakin banyak variasi pertanyaan yang didefinisikan maka akan sangat membantu sistem untuk menentukan jawaban yang tepat untuk dikirimkan sebagai jawaban pertanyaan. Gambar 2.5 menunjukkan cara pendefinisian pertanyaan didalam sebuah *intent*.



**Gambar 2.5 Pembuatan *Intent***

Gambar 2.5 di atas penulis mencoba membuat sebuah *intent* yang diberi nama “istilah.obstetri” , lalu setelah itu penulis membuat beberapa kalimat atau pertanyaan, yaitu diantaranya “apa itu obstetri”, ”pengertian obstetri”, ”obstetri”, “maksud obstetri?”. Kata atau kalimat tersebut merupakan perkiraan pertanyaan yang memiliki kemungkinan besar *end-user* atau pengguna tanyakan. *Intent* ini dibuat dengan tujuan untuk menampung segala pertanyaan yang berkaitan dengan obstetri, apabila *end-user* mengirim pertanyaan mengenai obstetri, dan apabila sesuai dengan pertanyaan yang sudah ada di basis data seperti di atas maka *Agent* akan dengan mudah untuk menentukan maksud dari pertanyaan tersebut dan dapat mengirim jawaban dengan tepat. Apabila pertanyaan sapaan yang dikirim belum ada di basis data maka *developer* dapat mendefinisikan ke dalam daftar tersebut.

Hal yang telah dijelaskan sebelumnya belum cukup untuk membuat *Agent* dapat memberikan jawaban yang tepat, namun untuk lebih memahami lebih lanjut tentang *platform* Dialogflow maka dari itu penulis akan menjabarkan istilah-istilah yang penulis pilih dan penting di dalam *platform* ini.

#### a. *Agents*

*Agents* dapat juga disamakan dengan sebuah modul. *Agents* inilah yang akan mengelola seluruh alur percakapan (*conversation flow*). Di dalam *Agents* ini terdapat sekumpulan *intents*, dan *entities*.

b. *Machine Learning*

*Machine learning* memungkinkan Agents untuk memahami input pengguna dalam bahasa alami dan mengubahnya menjadi data terstruktur, serta mengekstraksi parameter yang relevan. Dalam terminologi Dialogflow, *Agents* menggunakan algoritma *machine learning* untuk mencocokkan permintaan pengguna dengan maksud tertentu dan menggunakan entitas untuk mengekstrak data yang relevan. *Agents* belajar dari data yang telah diberikan sebelumnya serta dari model bahasa yang dikembangkan oleh Dialogflow. Berdasarkan data ini, *Agents* membangun sebuah model (algoritma) untuk membuat keputusan tentang maksud mana yang harus dipicu oleh input pengguna dan data apa yang perlu diekstraksi. Model menyesuaikan secara dinamis sesuai dengan perubahan yang dibuat di *Agents* dan di *platform* Dialogflow. Untuk memastikan bahwa modelnya membaik, *Agents* perlu terus dilatih pada *log* percakapan nyata.

c. *Intents*

*Intents* merupakan sebuah tempat untuk memetakan pertanyaan apa yang dikirim oleh pengguna dan tindakan apa yang harus dilakukan. Tujuan dari *intents* adalah mendefinisikan tata bahasa percakapan dan tugas apa yang harus dilakukan saat pengguna menggunakan frasa tertentu

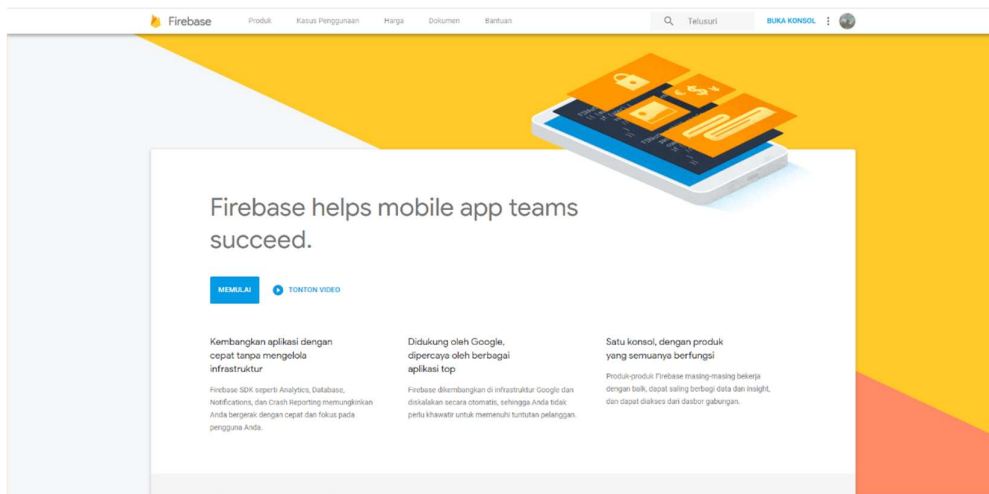
d. *Entities*

*Entities* adalah sebuah alat yang sangat kuat untuk mengidentifikasi nilai yang berguna dari sebuah masukan bahasa alami dalam hal ini adalah pertanyaan yang dikirim oleh pengguna. Di dalam *platform* Dialogflow terdapat 3 jenis *entities*, yaitu *System Entities*, *Developer Entities*, dan *User Entities*. *System Entities* merupakan *entities* yang sudah dibenamkan di dalam sistem, artinya *entities* tersebut sudah tersedia, *Developer Entities* merupakan *entities* yang dibuat dan didefinisikan oleh *developer*, *User Entities* adalah yang didefinisikan untuk setiap pengguna pada setiap permintaan.

## 2.4. Teknologi Firebase

Langkah-langkah menggunakan Firebase Authentication adalah sebagai berikut:

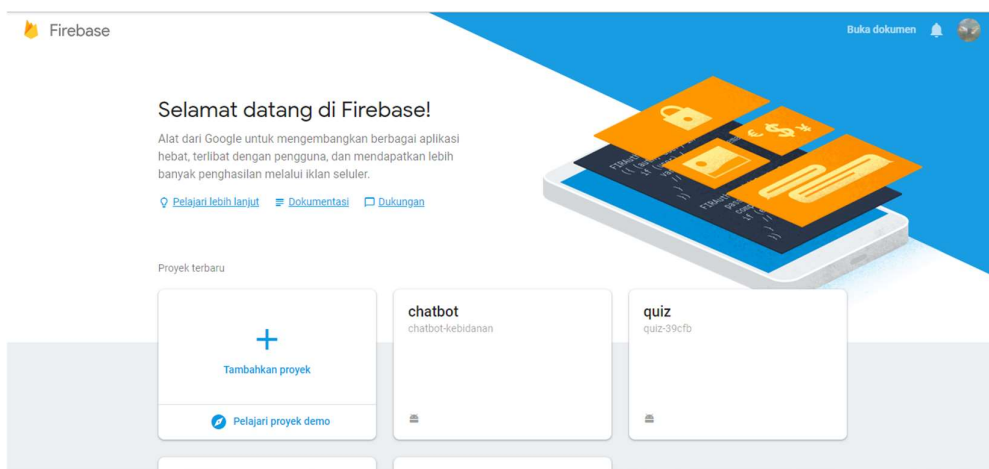
1. Pergi ke tautan <https://firebase.google.com/> seperti pada Gambar 2.6, kemudian pilih tombol Memulai dan melakukan login menggunakan akun google.



**Gambar 2.6 Halaman Awal Firebase**

### 2. Buat Proyek

- i. Pilih tombol Tambahkan proyek seperti terlihat pada Gambar 2.7 Halaman Awal Konsol Firebase.



**Gambar 2.7 Halaman Awal Konsol Firebase**

- ii. Masukkan *field* Nama Proyek dan Wilayah, ID Proyek akan secara otomatis terisi, kemudian pilih tombol Buat Proyek, lalu pilih tombol lanjutkan seperti pada Gambar 2.8 Mengisi Form Buat Proyek.

**Tambahkan project**

Nama proyek  
Midwify

ID project  
midwify

Lokasi Analytics  
Indonesia

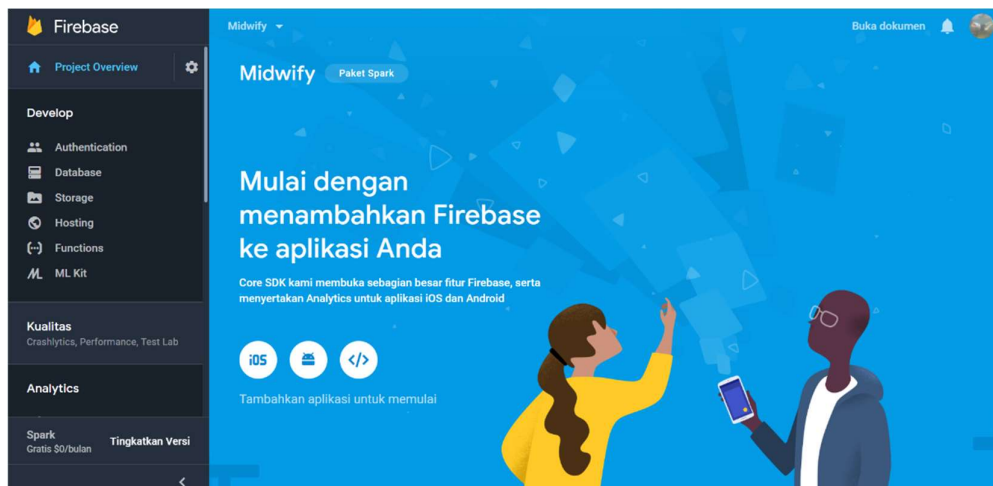
Lokasi Cloud Firestore  
us-central

Gunakan setelan default untuk membagikan data Google Analytics for Firebase

Tips: Project tersedia untuk aplikasi lintas platform

**Gambar 2.8 Mengisi *Form* Buat Proyek**

3. Tambahkan Firebase ke Aplikasi Android, dengan cara pilih tombol dengan logo Android seperti terlihat pada Gambar 2.9.



**Gambar 2.9 Halaman Dashboard Konsol Firebase**

- Mengisi *form* yang berisi nama aplikasi untuk menambahkan Firebase ke aplikasi Android seperti pada Gambar 2.10.

× Menambahkan Firebase ke aplikasi Android

1 Daftarkan aplikasi

Nama paket Android ⓘ  
com.azizhahkim.midwify

Nama aplikasi (opsional) ⓘ  
Freemium Android App

Sertifikat penandatanganan debug SHA-1 (opsional) ⓘ  
00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

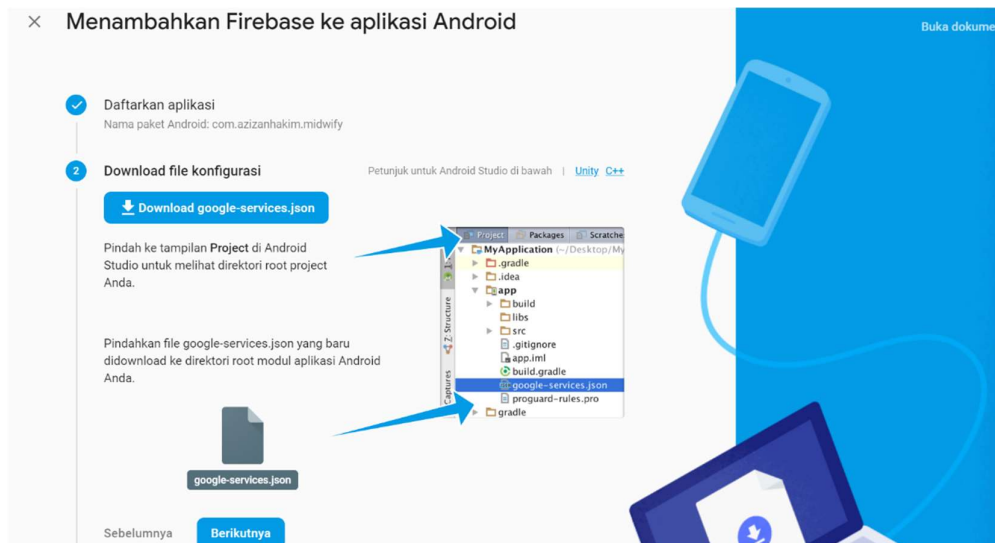
Diperlukan untuk Dynamic Links, Invites, dan Login dengan Google atau dukungan nomor telepon di Auth. Edit SHA-1 di Setelan.

Daftarkan aplikasi

Buka dokumen

**Gambar 2.10 Form Daftarkan Aplikasi Android**

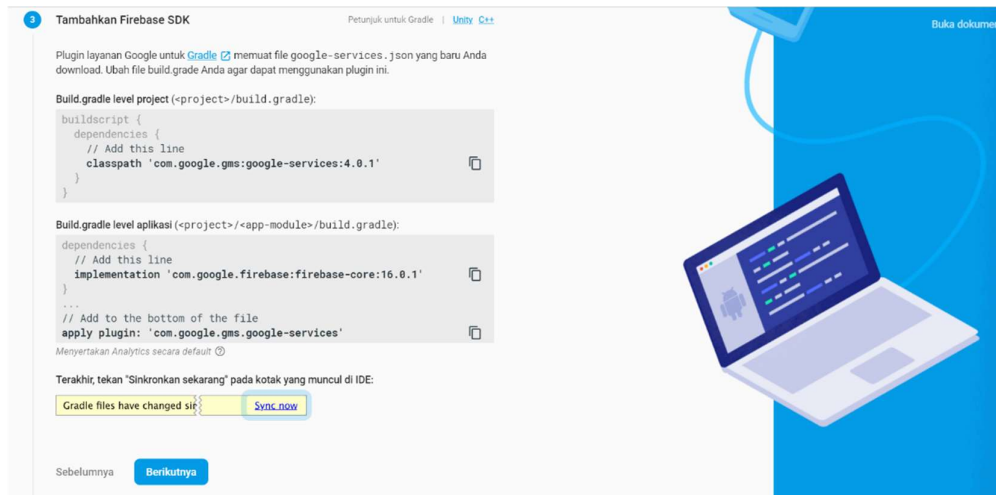
- Unduh dan simpan google-service.json di folder aplikasi Android seperti diterangkan pada Gambar 2.11.



**Gambar 2.11 Unduh Google-Service.Json**

- Ubah *file* build.grade agar dapat menggunakan *plugin* Firebase seperti Gambar 2.12.



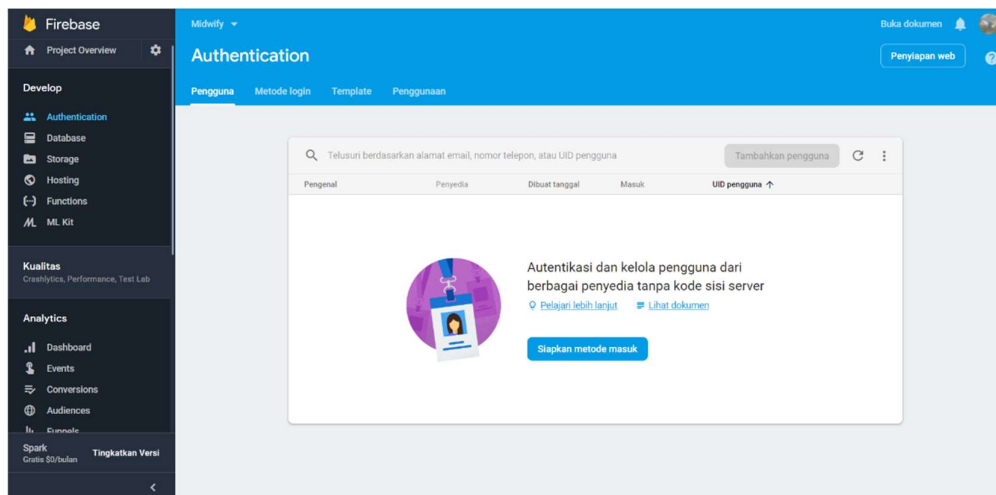


**Gambar 2.12 Tambahkan Firebase SDK**

## 7. Firebase Authentication

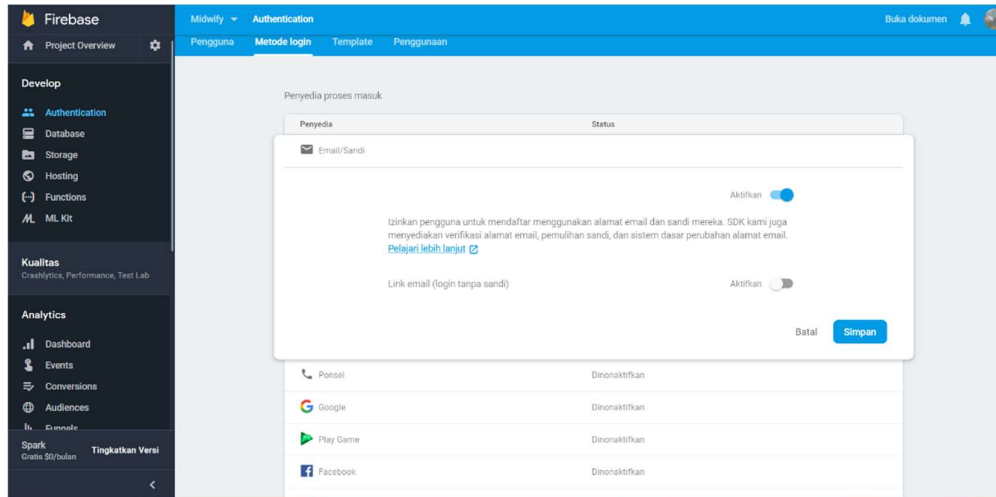
Firebase Authentication untuk memberikan autentikasi pada setiap pengguna yang akan menggunakan aplikasi *chatbot* Midwify. Berikut Langkah-langkah menggunakan Firebase Authentication adalah sebagai berikut:

- a. Pilih tombol Siapkan metode masuk untuk dapat menggunakan autentikasi dari Firebase Authentication seperti Gambar 2.13.



**Gambar 2.13 Halaman Awal Firebase Authentication**

- b. Pilih email pada menu Metode login untuk mengaktifkan autentikasi berdasarkan email pengguna seperti pada Gambar 2.14.

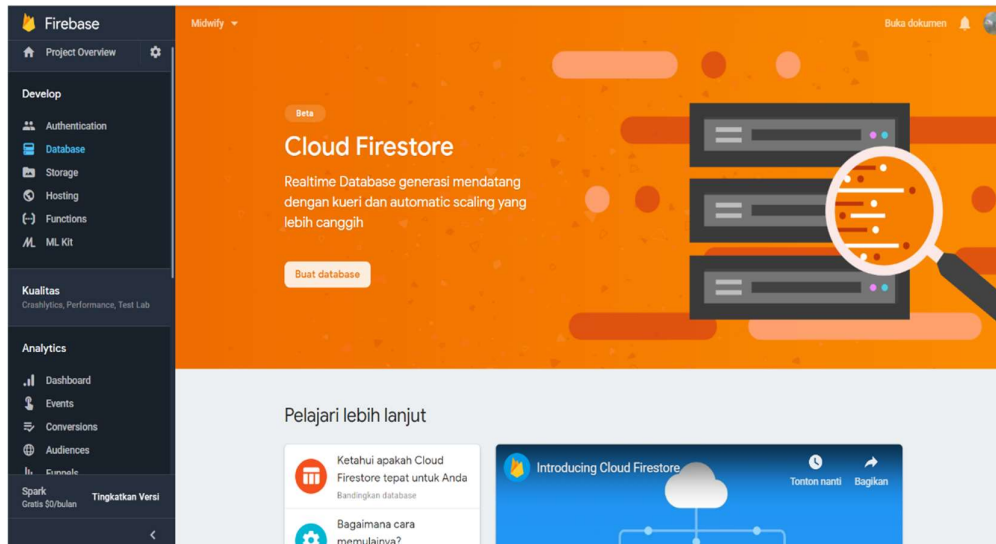


**Gambar 2.14 Halaman Metode Login**

## 8. Firebase Database

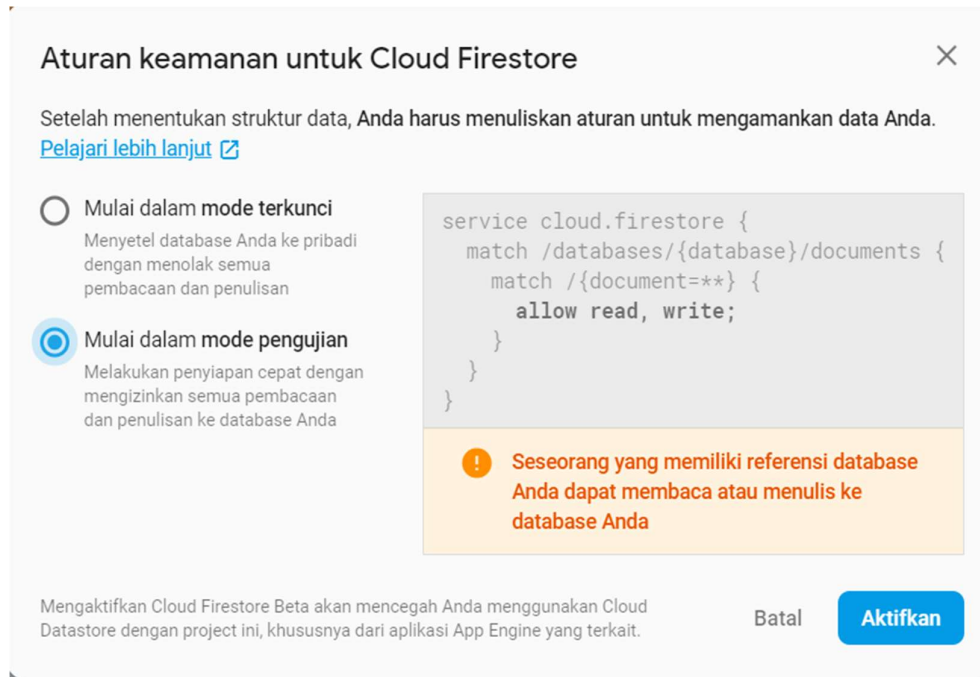
Firebase Database atau Cloud Firestore digunakan sebagai tempat penyimpanan Database Aplikasi Midwify. Berikut Langkah-langkah menggunakan Firebase Cloud Firestore.

- a. Pilih tombol Buat database untuk memulai membuat database yang dapat dilihat pada Gambar 2.15.



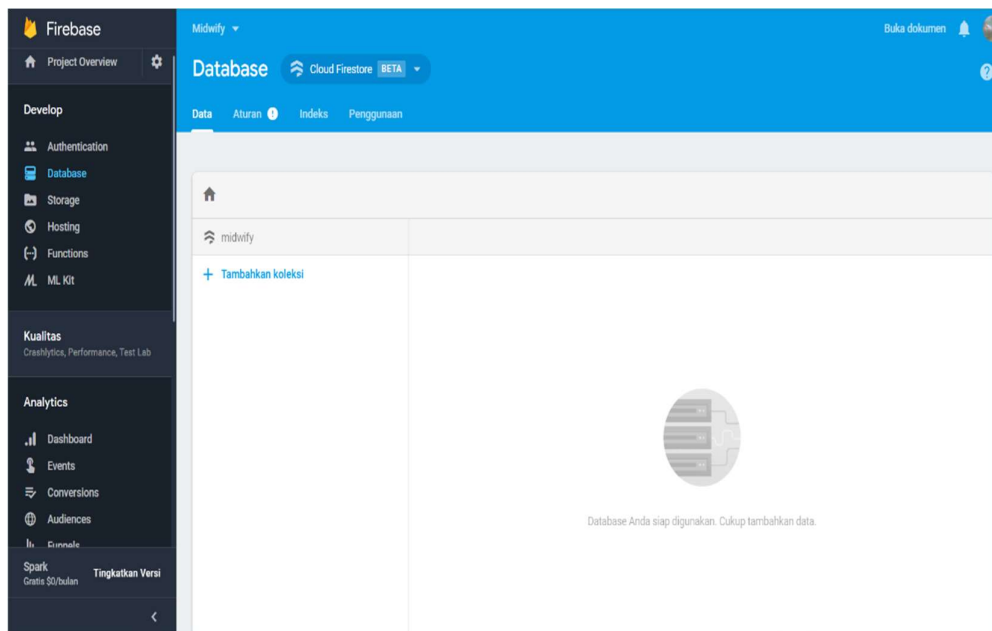
**Gambar 2.15 Halaman Awal Firebase Cloud Store**

- b. Pilih Mulai dalam mode pengujian dan kemudian pilih tombol Aktifkan seperti pada Gambar 2.16.



**Gambar 2.16 Atur keamanan Cloud Firestore**

- c. Pilih Tambahkan koleksi untuk memulai membuat Database yang dibutuhkan seperti pada Gambar 2.17.



**Gambar 2.17 Membuat Database**

## 2.5. Teknologi Google Speech-to-Text

*Google speech recognition* ini merupakan bagian dari *virtual keyboard input* yang telah disediakan oleh paket sistem Android dan dikenal dengan nama *Google voice typing*. *Google voice typing* dapat diatur melalui pengaturan di masing-masing perangkat *smartphone* Android, seperti pengaturan masukan Bahasa (*language input*). Untuk aplikasi *chatbot* Midwify, *language input* yang digunakan yaitu Bahasa Indonesia.

Secara *default*, Teknologi *Google Speech-to-Text* atau *Google Speech Recognition* ini telah termasuk di dalam paket android SDK. Untuk menggunakan *library* ini, dibutuhkan *Interface* yang dinamakan dengan *RecognizerIntent*. *RecognizerIntent* digunakan untuk mendapatkan hasil konversi berupa *String*.

## 2.6. Teknologi Dialogflow API

Teknologi *Dialogflow API (Application Programming Interface)* merupakan salah satu teknologi yang digunakan dalam pembangunan aplikasi *chatbot* midwify untuk menyimpan pengetahuan (*knowledge*) *bot* midwify. Pengetahuan ini berisi informasi percakapan mengenai istilah kebidanan dalam mata kuliah asuhan kebidanan 1 (kehamilan).

Untuk dapat memanfaatkan fitur *Dialogflow API*, diharuskan membuat *Agent* terlebih dahulu melalui *Dialogflow developer console*. *Agent* ini yang nantinya akan dilatih atau diperbarui basis pengetahuannya (*knowledge base*) agar dapat merespon setiap pesan pengguna dengan tepat. Semakin banyak dan beragam masukan terhadap basis pengetahuannya maka semakin cerdas *agent/bot* yang nantinya akan menjawab pesan pengguna. Sebelum pada tahap pembuatan *agent*, pertama-tama yang harus *developer* lakukan ialah mengunjungi tautan <https://console.dialogflow.com/> dan kemudian melakukan *login/sign up* dengan menggunakan akun Google seperti pada Gambar 2.18



Sign in with



In order to improve the system security, we are migrating authorization exclusively to Google accounts

[Terms of use and privacy policy](#)

### Gambar 2.18 Halaman Sign in Dialogflow Developer Console

Selanjutnya ditampilkan halaman *term of services* seperti pada Gambar 2.19 dan memilih negara asal dan mencentang *agreement* dari *term of services*, serta memilih tombol *Accept* atau terima.

Please review your account settings

Country or territory \*

Email preferences

Stay up-to-date with occasional emails from our team

- News and tips**  
Learn about new features, enhancements and tips
- Feedback and testing**  
Participate in surveys and pilots to improve Dialogflow

Terms of Service \*

Yes, I have read and accept the agreement.

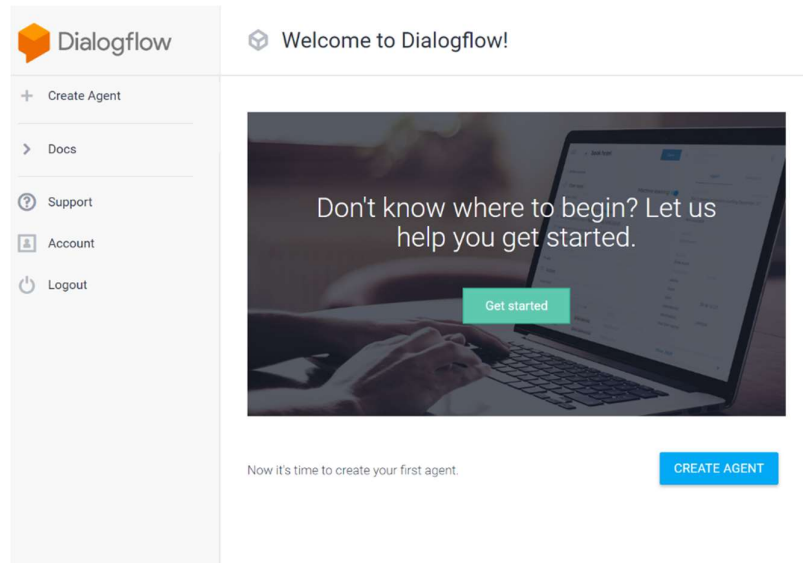
By proceeding and clicking the button below, you agree to adhere to the [Terms of Service](#).

Additionally, you may have access to certain Firebase services. You agree that your use of Firebase services will adhere to the applicable [Firebase Terms of Service](#). If you integrate any apps with Firebase on this project, by default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your Firebase settings at anytime.

ACCEPT

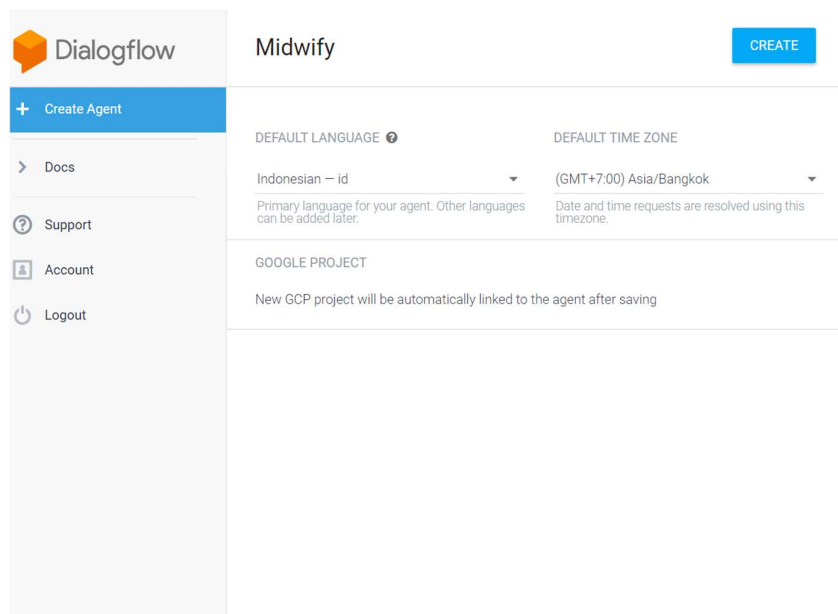
### Gambar 2.19 Halaman Review Akun

Setelah login, selanjutnya *developer* akan dialihkan pada halaman *developer console*. *Developer* dapat memilih tombol *Create Agent* Untuk membuat *agent* baru yang terletak di sisi kiri atas seperti pada Gambar 2.20.



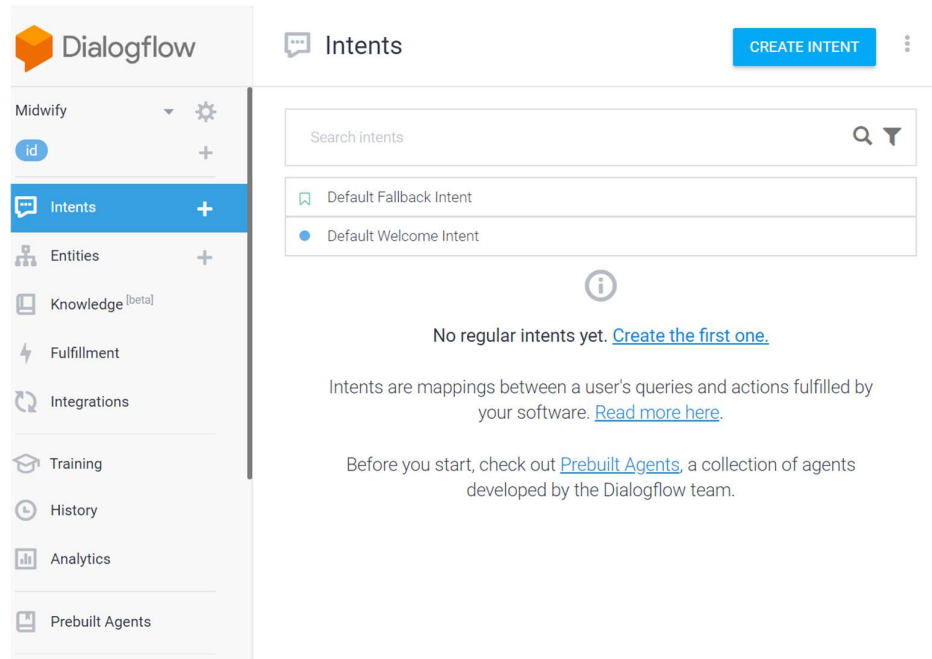
**Gambar 2.20 Halaman Dashboard Dialogflow**

Selanjutnya *developer* melakukan konfigurasi pada *Agent* yang akan dibuat. Konfigurasi tersebut terdiri dari nama *agent*, bahasa, dan zona waktu seperti terlihat pada Gambar 2.21.



**Gambar 2.21 Konfigurasi Agent**

Setelah *agent* berhasil dibuat, selanjutnya *developer* dapat menambahkan basis pengetahuan (*knowledge base*) kepada *agent*. Basis pengetahuan tersebut berupa *Intent*. Langkah selanjutnya bagi *developer* yaitu membuat *Intent*. Pilih menu *Intent* yang terdapat di sisi kiri dan pilih menu *Create Intent* yang dapat dilihat pada Gambar 2.22.



**Gambar 2.22 Membuat Intent**

Berikut ini adalah contoh pembuatan *Intent*, misalnya *intent* dengan nama istilah anemia. Intent terdiri dari frasa latih (*training phrases*) dan tanggapan (*responses*). Frasa latih dari istilah anemia dapat dilihat pada Gambar 2.23.

istilah.anemia SAVE

Contexts ? ∨

Events ? ∨

Training phrases ? Search training phrases 🔍 ∧

” add user expression

” apa yang dimaksud dengan anemia

” istilah anemia

” arti dari anemia

” pengertian anemia

” anemia

Action and parameters ∧

Enter action name

**Gambar 2.23 Frasa latih dari intent**

Berikut merupakan *responses* dari *intent* istilah.anemia yang dapat dilihat pada Gambar 2.24.

istilah.anemia SAVE

Action and parameters ∧

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

[+ New parameter](#)

Responses ? ∧

DEFAULT GOOGLE ASSISTANT +

Text response 🔍 🗑️

- kekurangan sel darah merah (eritrosit)
- Anemia dapat diketahui dengan mengecek HB ke laboratorium, atau dengan cara melihat bagian kelopak mata, jika berwarna putih pucat dapat dipastikan pasien tersebut mengalami anemia.
- Enter a text response variant ⌵

ADD RESPONSES

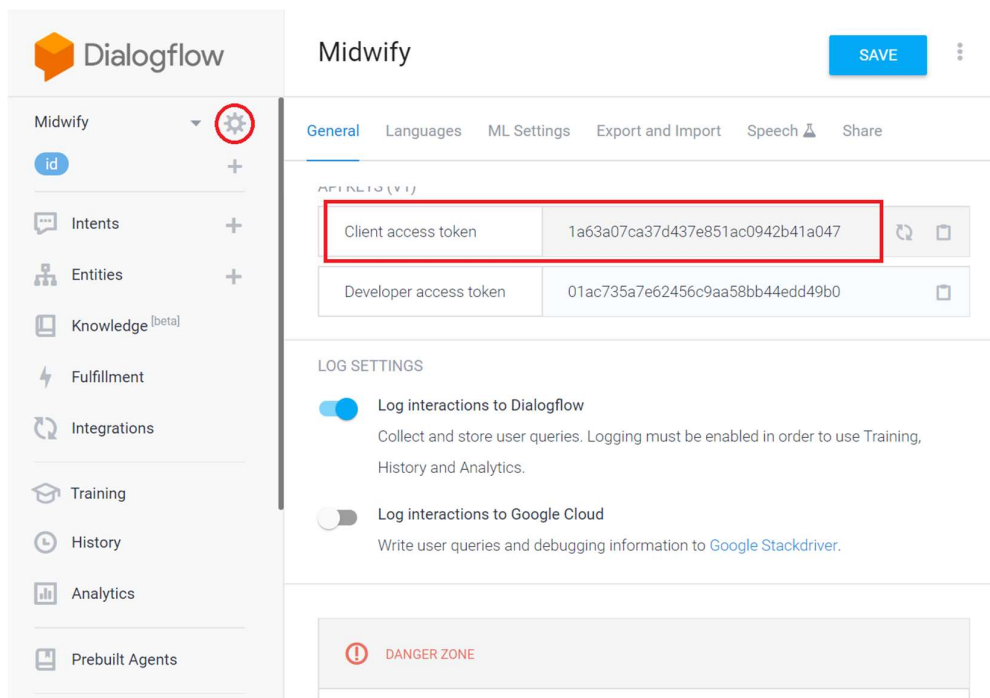
Set this intent as end of conversation ?

Fulfillment ? ∨

**Gambar 2.24 Responses dari Intent**



Pada kolom *Training Phrases* (Gambar 2.23) adalah semua kemungkinan frasa yang akan digunakan oleh pengguna dalam percakapan dengan *bot*. Sedangkan pada kolom *Responses* (gambar 2.24) adalah beberapa respon yang akan digunakan oleh *bot* dalam percakapan dengan pengguna. Kalimat-kalimat di dalam *training phrases* kemudian ditraining agar dapat menentukan similaritas terhadap kalimat yang akan digunakan oleh pengguna. Semakin banyak data *training phrases* maka hasil yang diperoleh akan semakin optimal. Jika kalimat dari pengguna cocok dengan *intent* istilah anemia, maka respon akan diberikan secara acak. Setelah *agent* siap, maka langkah selanjutnya ialah implementasi pada aplikasi. Namun sebelum itu, diperlukan token *client access* untuk inisialisasi. Token *client access* dapat diakses dengan memilih menu *setting*, kemudian memilih menu *general*, dan memilih *client access token* seperti pada Gambar 2.25.



**Gambar 2.25 Mendapatkan Token Client Access**

## 2.7. Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak 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 didirikan Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008. Logo Android dapat dilihat pada Gambar 2.26.



**Gambar 2.26 Logo Android**

Antarmuka pengguna Android umumnya berupa manipulasi langsung, menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser, mengetuk, dan mencubit untuk memanipulasi objek di layar, serta papan ketik virtual untuk menulis teks. Selain perangkat layar sentuh, Google juga telah mengembangkan Android TV untuk televisi, Android Auto untuk mobil, dan Android Wear untuk jam tangan. Masing-masing memiliki antarmuka pengguna yang berbeda. Varian Android juga digunakan pada komputer jingjing, komputer permainan, kamera digital, dan peralatan elektronik lainnya [9].

### 2.7.1. Fitur Android

Perangkat Android memiliki beberapa fitur perangkat keras di dalamnya. Inilah yang dapat di manfaatkan developer dalam membangun aplikasi. Diantaranya:

1. *Touchscreen*
2. *GPS*
3. *Accelerometer*
4. *SD Card*

Android memiliki banyak fitur perangkat lunak [9] yang dapat digunakan oleh pengembang (*developer*) dalam mengembangkan aplikasi, yaitu diantaranya:

1. Internet
2. Dukungan Audio dan Video
3. *Contact*
4. *Security*
5. Google APIs

#### **2.7.2. Versi Android**

Sistem operasi Android terdiri dari beberapa versi. Setiap versi Android terbaru memiliki nama-nama unik tersendiri yang terinspirasi dari nama-nama makanan penutup (*dessert*) yang terkenal di dunia. Tiap versi Android terbaru yang rilis memiliki beberapa jenis kelebihan mulai dari tampilan hingga optimasi keamanan. Berikut daftar nama versi dari Sistem Operasi Android [9], diantaranya adalah:

- a. Android versi 1.0 Alpha dirilis tanggal 23 September 2008.
- b. Android versi 1.1 Beta dirilis tanggal 09 Februari 2009.
- c. Android versi 1.5 Cupcake dirilis tanggal 30 April 2009.
- d. Android versi 1.6 Donut dirilis tanggal 15 September 2009.
- e. Android versi 2.0 Eclair dirilis tanggal 26 Oktober 2009.
- f. Android versi 2.2 Froyo dirilis tanggal 10 Mei 2010.
- g. Android versi 2.3 Gingerbread dirilis tanggal 06 Desember 2010.
- h. Android versi 3.0 Honeycomb dirilis tanggal 22 Februari 2011.
- i. Android versi 4.0 Ice Cream Sandwich tanggal 19 Oktober 2011.
- j. Android versi 4.1 Jelly Bean dirilis tanggal 09 Juli 2012.
- k. Android versi 4.4 Kitkat dirilis tanggal 31 Oktober 2013.
- l. Android versi 5.0 Lollipop dirilis tanggal 17 Oktober 2014.
- m. Android versi 6.0 Marshmallow dirilis tanggal 28 Mei 2015.

- n. Android versi 7.0 Nougat dirilis tanggal 22 Agustus 2016.
- o. Android versi 8.0 Oreo dirilis tanggal 21 Agustus 2017.
- p. Android versi 9.0 Pie dirilis tanggal 6 Agustus 2018.

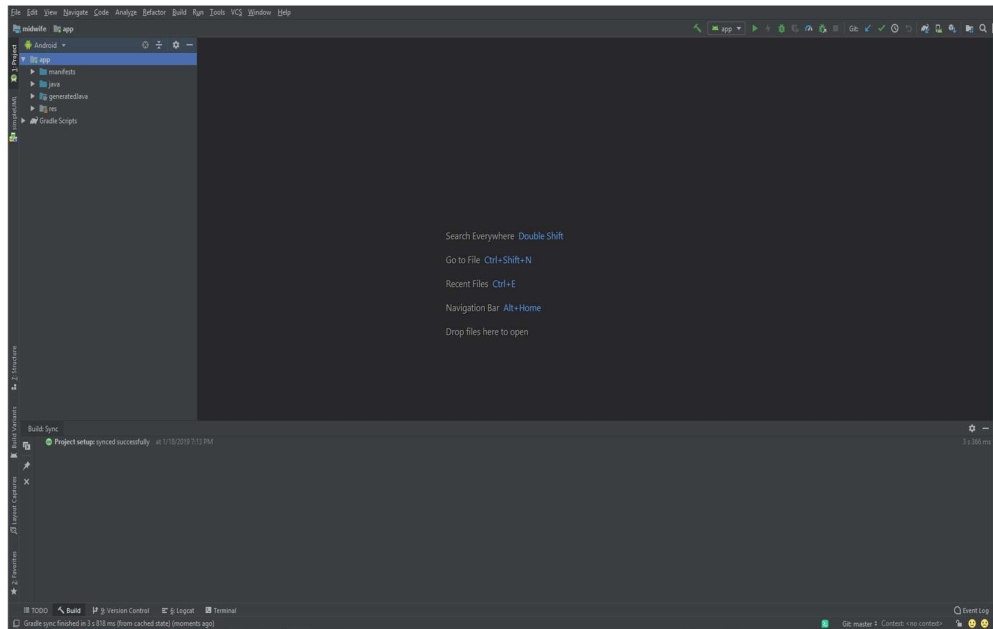
### 2.7.3. Android Studio

Android studio merupakan sebuah software tools *Integrated Development Environment* (IDE) untuk platform Android. Android studio ini diluncurkan pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Manajer Produk dari Google, Ellie Powers. Logo Android Studio dapat dilihat pada Gambar 2.27.



**Gambar 2.27 Logo Android Studio**

Android studio ini bersifat gratis dan dibawah *Apache License 2.0*. Android Studio awalnya dimulai dengan versi 0.1 pada bulan mei 2013, Kemudian dibuat versi beta 0.8 yang dirilis pada bulan juni 2014, yang paling terbaru dirilis versi 3.3 pada bulan Desember 2018. Android studio berbasiskan aplikasi JetBrains IntelliJ IDEA, yang kemudian Android Studio didesain khusus untuk *Android Development*. Untuk saat ini, Android Studio sudah dapat digunakan pada sistem operasi Windows, Mac OS X, dan Linux. Lokasi *url* untuk mengunduh aplikasi Android Studio yaitu <https://developer.Android.com/studio/index.html?hl=id> [9]. Tampilan Android Studio versi 3.3 dapat dilihat pada Gambar 2.28.



**Gambar 2.28 Tampilan Awal Android Studio**

### **Syarat Minimum Perangkat Keras:**

#### **Windows:**

1. Microsoft Windows 7/8/10 (32 atau 64 bit)
2. RAM minimum 3 GB, RAM yang digunakan disarankan 8 GB; ditambah 1 GB untuk Emulator Android
3. Ruang Disk minimum yang tersedia 2 GB. Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator)
4. Resolusi layar minimum 1280 x 800

#### **Mac:**

1. Mac OS X 10.10 (Yosemite) atau lebih tinggi, hingga 10.12 (mac OS Sierra).
2. RAM minimum 3 GB, RAM yang disarankan 8 GB; ditambah 1 GB untuk Emulator Android
3. Ruang Disk minimum yang tersedia 2 GB. Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator)
4. Resolusi layar minimum 1280 x 800

Linux:

1. Desktop GNOME atau KDE. Telah diuji pada Ubuntu 12.04, Pricise Pangolin (distribusi 64-bit dan 32-bit).
2. Distribusi 64-bit yang mampu menjalankan aplikasi 32-bit.
3. GNU C Library (glibc) 2.19 atau yang lebih baru.
4. RAM minimum 3 GB, RAM yang disarankan 8 GB; dtambah 1 GB untuk Emulator Android.
5. Ruang Disk minimum yang tersedia 2 GB. Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator).
6. Resolusi layar mininum 1280 x 800.

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

Salah satu persyaratan dari Android Studio untuk dapat mengembangkan aplikasi Android adalah JDK atau *Java Development Kit* [9]. Maka dari itu, langkah pertama yang harus dilakukan sebelum menggunakan Android Studio adalah menginstalasi JDK. Java Development Kit ini dapat diunduh pada laman *web* <http://www.oracle.com/technetwork/java/javase/download/jdk8-downloads-2133151.html>.

#### **2.7.5. Emulator**

Untuk mencoba program yang dibuat di Android Studio, diperlukan *emulator* Android. Atau dapat pula langsung dijalankan menggunakan *smartphone* Android. Perlu diketahui bahwa menjalankan program Android melalui *emulator* membutuhkan RAM komputer yang lebih besar dibandingkan menggunakan *smartphone* secara langsung. Sebaiknya komputer memiliki minimal RAM 8 GB atau lebih jika ingin menggunakan *emulator*. Ada Beberapa software emulator Android yang dapat digunakan misalnya *Android Virtual Device* (AVD) yang disediakan Android SDK atau *emulator* yang disediakan pihak ketiga seperti Genymotion, Nox, dan Bluestack [9].

## 2.8. Multimedia

Konsep multimedia sering berubah-ubah dari waktu ke waktu dengan mengikuti perkembangan zaman. Pada tahun 1960-an multimedia dimaknai sebagai gabungan atau kumpulan dari berbagai peralatan media yang bermacam-macam untuk digunakan pada saat presentasi. Pada tahun 1990-an, multimedia diartikan sebagai mentransmisikan teks, audio, dan gambar dengan *real time*. Namun di masa sekarang, multimedia dimaknai sebagai suatu sistem komunikasi interaktif berbasis komputer yang mampu menciptakan, menyimpan, menyajikan, dan mengakses kembali informasi berupa teks, grafik, suara, video atau animasi [10].

Istilah multimedia terlebih dahulu dikenal dalam seni pertunjukan teater. Pertunjukan multimedia di dalam seni pertunjukan teater memanfaatkan lebih dari satu medium. Pertunjukan bermultimedia tersebut mencakup media monitor video, *synthesized band*, dan karya seni manusia sebagai bagian dari pertunjukan. Dalam perkembangan selanjutnya penggunaan istilah multimedia mengalami sedikit perubahan, yang tadinya memiliki makna seni pertunjukan dengan menggunakan berbagai media beralih ke istilah komputer, maka akhirnya istilah multimedia masuk dalam kosa kata komputer.

Multimedia dalam konteks komputer menurut Hofstetter, yaitu penggunaan komputer untuk menyajikan dan menggabungkan teks, suara, gambar, animasi dan video dengan dan koneksi (*link*) alat bantu (*tools*) sehingga pengguna dapat melakukan navigasi, berinteraksi, berkarya dan berkomunikasi [11]. Berdasarkan pengertian itu, multimedia terdiri dari empat faktor, yaitu: (i) ada komputer yang mengkoordinasikan apa yang dilihat dan didengar, (ii) ada link yang menghubungkan pengguna dengan informasi, (iii) ada alat navigasi yang membantu pengguna menjelajah jaringan informasi yang saling terhubung, dan (iv) multimedia menyediakan tempat kepada pengguna untuk mengumpulkan, memproses, dan mengkomunikasikan informasi dengan ide secara interaktif [10].

Multimedia dapat dibagi dua berdasarkan kepada cara pengguna dalam berinteraksi dengan aplikasi multimedia, yaitu multimedia linear dan multimedia nonlinear. Multimedia linear bersifat tidak interaktif karena pengguna tidak memiliki kebebasan dalam menggunakan aplikasi, karena aplikasi multimedia yang digunakan tidak dilengkapi alat pengontrol sehingga akan menyajikan konten mulai dari awal hingga akhir dan akan mengulanginya dari awal kembali untuk penggunaan selanjutnya. Contohnya adalah film, animasi, dan videoklip. Sedangkan multimedia nonlinear bersifat interaktif karena pengguna memiliki kebebasan dalam menggunakan aplikasi, karena pengguna bebas menentukan apapun yang dikehendaki dari aplikasi untuk ditampilkan dengan alat pengontrol yang tersedia. Contohnya yaitu multimedia pembelajaran interaktif, laman web dan permainan komputer.

### **2.8.1. Multimedia Interaktif**

Multimedia berasal dari kata multi dan media. Secara etimologis Multi berasal dari bahasa Latin, yaitu *nouns* yang berarti banyak atau bermacam-macam. Sedangkan kata media berasal dari bahasa Latin, yaitu *medium* yang berarti perantara atau sesuatu yang dipakai untuk menghantarkan, menyampaikan, atau membawa sesuatu. Kata medium dalam *American Heritage Electronic Dictionary* diartikan sebagai alat untuk mendistribusikan dan mempresentasikan informasi [12]. Dapat disimpulkan bahwa multimedia merupakan perpaduan antara berbagai media (format file) yang berupa teks, gambar (vektor atau bitmap), grafik, sound, animasi, video, interaksi, dan sebagainya yang telah dikemas menjadi *file* digital (komputerisasi) serta digunakan untuk menyampaikan atau menghantarkan pesan kepada publik [10]. Karakteristik terpenting dari multimedia interaktif adalah pengguna tidak hanya memperhatikan media atau objek saja, melainkan juga dituntut untuk berinteraksi selama berinteraksi dengan aplikasi.



### 2.8.2. Elemen Multimedia

Multimedia interaktif mensinergikan atau menggabungkan berbagai media yang terdiri dari: a) teks; b) grafik; c) audio; dan d) interaktivitas [13].

#### a. Teks

Teks adalah suatu kombinasi huruf yang membentuk satu kata atau kalimat yang dapat menjelaskan suatu maksud atau materi pembelajaran yang dapat dipahami oleh pengguna yang membacanya. Teks memiliki berbagai macam jenis bentuk atau tipe (sebagai contoh: Time New Roman, Arial, Comic San MS), berbagai macam ukuran dan warna. Satuan dari ukuran suatu teks terdiri dari length dan size. Length biasanya menyatakan banyaknya teks dalam sebuah kata atau halaman. Size menyatakan ukuran besar atau kecil suatu huruf. Standar teks memiliki size 10 atau 12 poin. Semakin besar size suatu huruf maka semakin tampak besar ukuran huruf tersebut.

#### b. Grafik

Grafik adalah suatu medium berbasis visual. Seluruh gambar dua dimensi adalah grafik. Apabila gambar di render dalam bentuk tiga dimensi (3D), biasanya tetap disajikan melalui medium dua dimensi. Hal ini termasuk gambar yang disajikan lewat kertas, televisi ataupun layar monitor. Grafik bisa saja menyajikan kenyataan (realitas) atau hanya berbentuk ikonik. Contoh grafik yang menyajikan kenyataan adalah foto, dan contoh grafik yang berbentuk ikonik adalah kartun seperti gambar yang biasa dipasang dipintu toilet untuk membedakan toilet laki-laki dan perempuan.

#### c. Audio

Audio didefinisikan sebagai macam-macam bunyi dalam bentuk digital seperti suara, musik, narasi, dan sebagainya yang dapat didengar oleh pengguna untuk keperluan suara latar, penyampaian berbagai macam pesan yang dapat disesuaikan berdasarkan situasi penggunaannya. Penggunaan audio pada multimedia dapat berupa lagu, narasi, dan efek suara. Biasanya narasi ditampilkan bersama-sama dengan foto atau teks yang berguna untuk memperjelas informasi yang akan disampaikan.

d. Video

Video pada dasarnya adalah alat atau media yang dapat menunjukkan simulasi benda nyata. Angew dan Kellerman mendefinisikan video sebagai media digital yang menunjukkan susunan atau urutan gambar-gambar yang bergerak dan dapat memberikan ilusi/fantasi. Video juga sebagai sarana untuk menyampaikan informasi yang menarik, langsung, dan efektif. Video pada multimedia digunakan untuk menggambarkan suatu kegiatan atau aksi yang dilakukan [14].

e. Animasi

Animasi adalah suatu tampilan yang menggabungkan antara media teks, grafik, dan suara dalam suatu aktivitas pergerakan. Dalam multimedia, animasi merupakan penggunaan komputer untuk menciptakan gerak pada layer. Animasi dapat juga digunakan untuk menjelaskan serta mensimulasikan sesuatu yang sulit dilakukan oleh video.

f. Interaktivitas

Elemen ini sangat penting dalam multimedia interaktif. Elemen lain seperti teks, suara, foto, hingga video dapat disajikan oleh berbagai macam perangkat media seperti televisi dan VCD player, namun elemen interaktivitas hanya dapat disajikan oleh komputer. Elemen ini sangat memanfaatkan kemampuan komputer sepenuhnya. Aspek interaktif pada multimedia dapat berupa navigasi, simulasi, permainan, dan latihan. Apabila dalam suatu aplikasi multimedia, pengguna multimedia diberikan suatu kemampuan untuk mengontrol elemen-elemen yang tersedia, maka multimedia itu disebut dengan multimedia interaktif. Apabila dalam aplikasi multimedia disediakan struktur dari elemen terhubung yang dapat dikendalikan oleh pengguna, maka multimedia interaktif tersebut dikenal dengan *Hypermedia*.

## 2.9. Unified Modeling Language (UML)

*Unified Modeling Language* hanya merupakan suatu bahasa pemodelan visual yang. Oleh karena itu penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada umumnya UML lebih banyak digunakan pada metodologi berorientasi objek. UML pertama kali dikenalkan oleh Jim Rumbaugh, Ivars Jacobson, dan Grady Booch pada pertengahan tahun 1990. Berikut adalah jenis-jenis dari diagram UML [15]:

1. *Use case diagram*
  - a. Menggambarkan bagaimana sistem digunakan,
  - b. Mendeskripsikan apa yang sistem akan lakukan tanpa mendeskripsikan bagaimana sistem dapat menyelesaikannya,
  - c. Dibuat berdasarkan interaksi dan relasi dari individual *use case*,
  - d. Berisi aktor (*actor*), *event*, dan *use cases*, dan
  - e. Titik awal dari suatu pemodelan UML.

2. *Use case scenario*

Sebuah artikulasi verbal pengecualian terhadap perilaku utama yang dijelaskan oleh kasus penggunaan utama. *Use case scenario* adalah hasil dari instansiasi setiap *use case*. *Use case scenario* terbagi atas tiga bagian, diantaranya:

- a. Identifikasi dan inisiasi,
- b. *Step performed*, serta
- c. Kondisi, asumsi, dan pertanyaan.

3. *Activity diagram*

*Activity diagram* yaitu suatu diagram yang menggambarkan konsep aliran data/kontrol, aksi yang terstruktur, dan dirancang dengan baik dalam suatu sistem [15]. Berikut ini merupakan komponen dari *activity diagram* menurut Bock dalam *Journal of Object Technology* [15], diantaranya:

- a. *Activity node*

*Activity node* menggambarkan bentuk notasi dari beberapa proses yang beroperasi dalam kontrol dan nilai data.

b. *Activity edge*

*Activity edge* menggambarkan bentuk *edge* yang menghubungkan aliran aksi secara langsung, dimana menghubungkan input dan output dari aksi tersebut.

c. *Initial state*

Bentuk lingkaran berisi penuh melambangkan awal dari suatu proses.

d. *Decision*

Bentuk wajib dengan suatu *flow* yang masuk beserta dua atau lebih *activity node* yang keluar. *Activity node* yang keluar ditandai untuk mengindikasikan beberapa kondisi.

e. *Fork*

Satu *bar* hitam dengan satu *activity node* yang masuk beserta dua atau lebih *activity node* yang keluar.

f. *Join*

Satu *bar* hitam dengan dua atau lebih *activity node* yang masuk beserta satu *activity node* yang keluar, tercatat pada akhir dari proses secara bersamaan. Semua *actions* yang menuju *join* harus lengkap sebelum proses dapat berlanjut.

g. *Final state*

Bentuk lingkaran berisi penuh yang berada di dalam lingkaran kosong, menunjukkan akhir dari suatu proses.

4. *Sequence diagram*

*Sequence diagram* adalah sebuah diagram yang menggambarkan kolaborasi dari berbagai objek yang saling berinteraksi antar elemen dari suatu *class* [10]. Berikut ini merupakan komponen dalam *sequence diagram*, diantaranya adalah:

a. *Activations*

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

b. *Actor*

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

c. *Collaboration boundary*

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

d. *Parallel vertical lines*

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

e. *Processes*

*Processes* menjelaskan tentang tindakan/aksi yang dilakukan oleh *actor* dalam suatu waktu

f. *Window*

*Window* menjelaskan mengenai halaman yang tengah ditampilkan dalam suatu proses.

g. *Loop*

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

5. *Class diagram*

*Class diagram* adalah sebuah diagram yang menunjukkan hubungan antar *class* yang di dalamnya terdapat atribut dan fungsi dari suatu objek [15]. Pada *class diagram* terdapat tiga relasi dalam hal penggunaannya, diantaranya:

a. *Association*

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

b. *Generalization*

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

c. *Constraint*

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

## 2.10. Metode Pengujian Perangkat Lunak

Metode pengujian adalah suatu teknik menguji perangkat lunak, mempunyai mekanisme untuk menentukan data uji yang dapat menguji perangkat lunak secara lengkap dan mempunyai kemungkinan tinggi untuk menemukan kesalahan. Perangkat lunak dapat diuji dengan dua cara, yaitu:

### a. Pengujian *White Box*

Menurut Roger. S. Pressman, pengujian *white box* adalah metode desain uji kasus (*test case design*) yang menggunakan struktur kontrol desain prosedural untuk memperoleh uji kasus (*test case*) [16]. Teknik pengujian ini merupakan suatu Metode basis *path* yang digunakan untuk menentukan ukuran kompleksitas logika dari suatu logika. Metode *white box* yang digunakan berguna sebagai:

1. Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali dalam pengujian,
2. Mengukur kompleksitas logika dari desain prosedur dan sekaligus menjadi pedoman untuk mendapatkan konsisten jalur aplikasi,
3. Mengeksekusi semua perulangan (*looping*) pada batasan tertentu, dan
4. Menggunakan struktur data internal yang menjamin validitasnya.

### b. Pengujian *Black Box*

Pengujian *black box* merupakan pendekatan komplementer dari teknik pengujian *white box*, pengujian *black box* ini diharapkan mampu mengungkapkan kesalahan yang lebih luas dibandingkan dengan teknik pengujian *white box* . Pengujian *black box* berfokus pada pengujian persyaratan fungsional dari sistem, untuk mendapatkan serangkaian kondisi input yang sesuai dengan persyaratan fungsional suatu program. Uji coba *black box* bukan merupakan alternatif dari uji coba *white box*, tetapi merupakan suatu pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white box* [16]. Pada pengujian ini untuk menemukan kesalahan dalam beberapa kategori, diantaranya:

1. Fungsi-fungsi yang salah atau hilang,
2. Kesalahan interface.
3. Kesalahan dalam struktur data atau akses database eksternal.

4. Kesalahan performa.
5. Kesalahan inisialisasi dan terminasi.

Berbeda halnya dengan metode *white box* yang dilaksanakannya pada awal proses, uji coba *black box* diaplikasikan di beberapa tahapan berikutnya. Karena uji coba *black box* ini mengabaikan struktur kontrol, sehingga perintahnya hanya difokuskan pada informasi domain.

Menurut Srinivas Nidhra, keuntungan dari pengujian *black box* yaitu penguji tidak perlu memiliki pengetahuan mengenai Bahasa pemrograman tertentu. Tidak hanya bahasa pemrograman namun juga pengetahuan tentang pengaplikasiannya. Pada pengujian *black box*, *programmer* dan penguji (*tester*) bersifat independent satu sama lain.

Keuntungan lainnya bahwa pengujian dilakukan dari sudut pandang pengguna sehingga hal ini membantu untuk mengekspos setiap ambiguitas atau inkonsistensi dalam persyaratan spesifikasi [17]

