



# Android Quiz Bot *with* Dialogflow

*created by : Ade Guntur R*

## Intoduction Dialogflow

Pada kesempatan kali ini penulis akan berbagi cara untuk membuat Quiz Bot Dialogflow pada android. Namun sebelum kita praktek membuatnya kita harus mengenal apa itu Dialogflow?



(Gambar 1.1 Api.ai To Dialogflow)

Dialogflow dari Google (sebelumnya dikenal sebagai API.ai) adalah platform hebat untuk mengembangkan chatbot untuk banyak platform. Ada cukup tutorial dan dokumentasi yang tersedia bagi orang baru untuk mulai belajar dan menggunakan platform untuk membuat chatbot. Namun, dukungan untuk Android dan iOS tidak terdokumentasi dengan baik. Sebelumnya ketika Dialogflow dikenal sebagai API.ai, mereka memiliki *client library* Android dan iOS untuk mengintegrasikan chatbot ke dalam *mobile apps*. Tetapi setelah bergabung ke Google, *library* yang ada tidak diperbarui, penulis melihat Dialogflow ditambahkan ke platform Google Cloud dan API client berbasis Java baru sedang dikembangkan.

Dialogflow memungkinkan Anda membangun *conversational interfaces* diatas produk dan layanan Anda dengan menyediakan mesin *natural language understanding* (NLU) yang kuat untuk memproses dan memahami input bahasa alami. Modul ini membahas cara kerja Dialogflow dan bagaimana hal itu dapat membantu anda membuat *conversational interfaces* yang menyenangkan pengguna. *Traditional computer interface* membutuhkan input terstruktur dan dapat diprediksi untuk berfungsi dengan baik, yang membuat penggunaan *interface* ini tidak alami dan terkadang sulit. Jika pengguna tidak dapat dengan mudah mengetahui input terstruktur ini, mereka akan kesulitan menentukan apa yang harus dilakukan.

Misalnya, pertimbangkan permintaan pengguna yang mudah seperti "Seperti apa ramalan cuaca hari ini?". Pengguna lain mungkin juga bertanya:

"Seperti apa cuaca saat ini?"

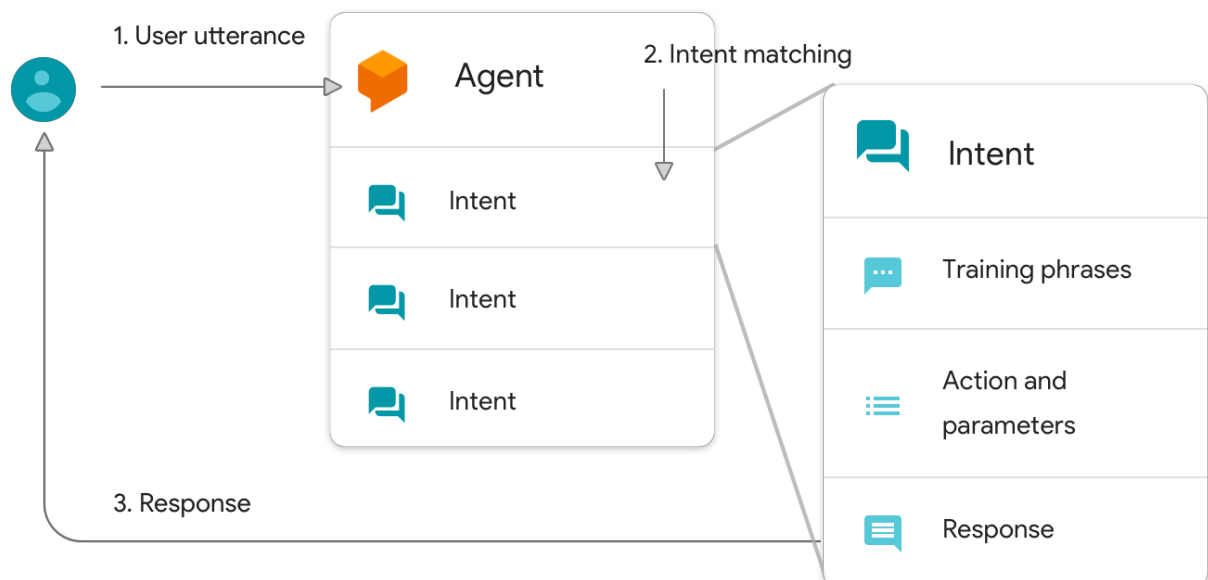
"Seperti apa suhu di Surabaya besok?"

"Seperti apa cuaca di tanggal 21?"

Bahkan dengan pertanyaan sederhana ini, Anda dapat melihat bahwa pengalaman percakapan sulit untuk diimplementasikan. Menafsirkan dan mengolah bahasa alami membutuhkan pengurai bahasa yang sangat kuat yang mampu memahami nuansa bahasa.

Kode anda harus menangani semua jenis permintaan yang berbeda ini (dan kemungkinan banyak lagi) untuk melakukan logika yang sama: mencari beberapa informasi perkiraan untuk waktu dan lokasi. Karena alasan ini, *Traditional computer interface* cenderung memaksa pengguna untuk memasukkan permintaan standar yang terkenal dan merugikan pengalaman pengguna, karena itu hanya lebih mudah.

Namun, Dialogflow memungkinkan anda dengan mudah mencapai pengalaman percakapan pengguna dengan menangani pemahaman *Natural language understanding* (NLU) untuk Anda. Ketika Anda menggunakan Dialogflow, Anda membuat *Agent* yang dapat memahami nuansa bahasa manusia yang luas dan beragam dan menerjemahkannya ke makna standar dan terstruktur yang dapat dipahami oleh aplikasi dan layanan Anda. Mari kita lihat bagaimana Dialogflow dapat menangani contoh sebelumnya untuk permintaan ramalan cuaca.

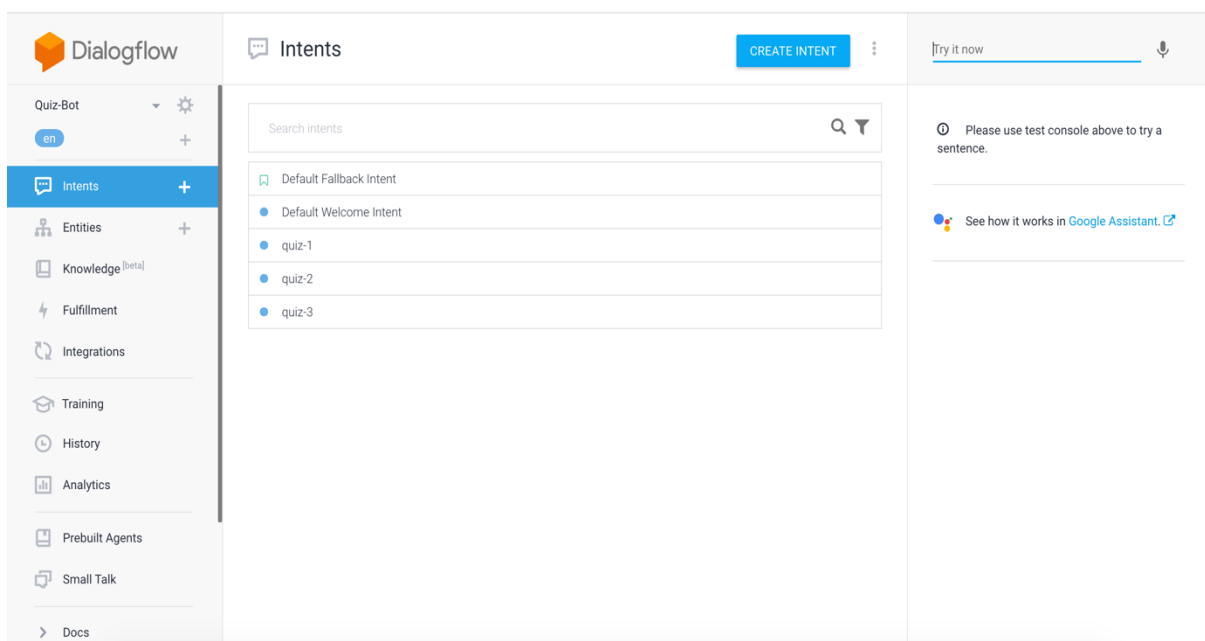


(Gambar 1.2 Contoh bagaimana Dialogflow menangani ucapan pengguna)

Untuk mencari ramalan cuaca, Anda mungkin memerlukan beberapa informasi, seperti waktu yang diinginkan pengguna untuk ramalan dan lokasi mereka. Namun, seperti yang kami sebutkan sebelumnya, pengguna yang berbeda mungkin meminta perkiraan dengan cara yang berbeda. Dialogflow dapat memahami perbedaan-perbedaan ini dan menerjemahkannya ke maksud pengguna standar untuk mendapatkan perkiraan. Ini kemudian dapat menguraikan permintaan pengguna untuk data terkait yang Anda butuhkan untuk memenuhi permintaan tersebut. Dalam hal ini, itulah waktu dan lokasi yang diinginkan pengguna untuk ramalan cuaca. Terakhir, Anda dapat menggunakan data ini untuk mencari cuaca dengan *REST API* publik dan mengembalikan cuaca kepada pengguna dalam bentuk respons.

## Building blocks of Dialogflow

Berikut ini adalah tampilan pada website <http://console.dialogflow.com/> yang dimana tampilan tersebut adalah blok bangunan dialogflow yang nantinya akan kita gunakan.



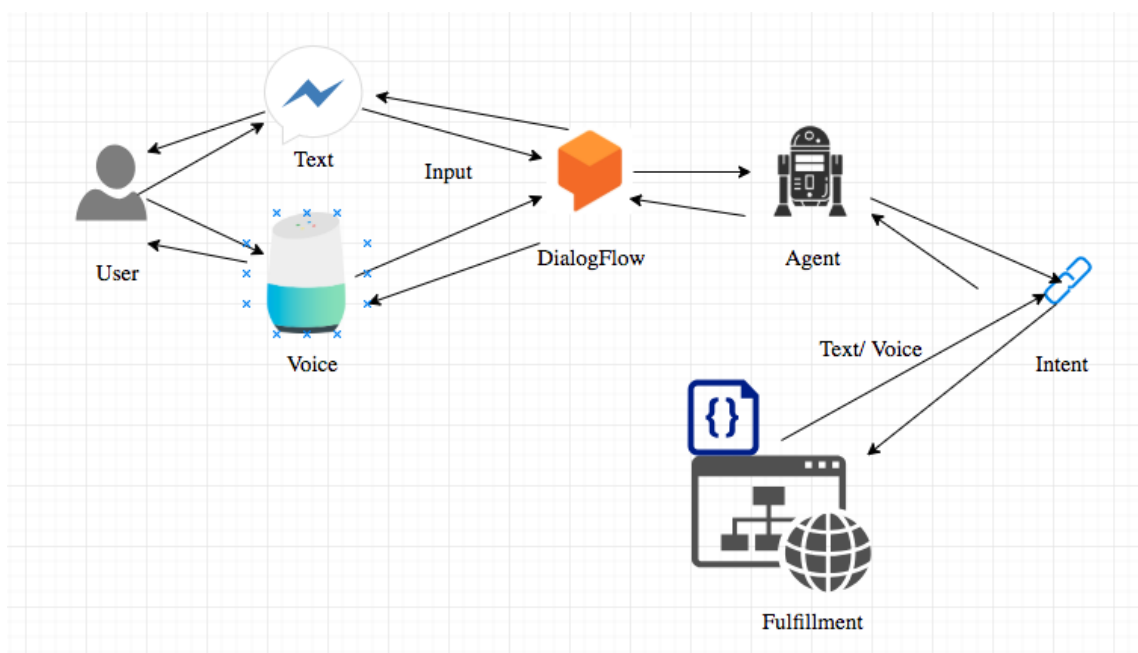
(Gambar 1.3 Tampilan website Dialogflow)

Penjelasan mengenai masing – masing blok sebagai berikut :

1. **Agents** : Membantu mengonversi permintaan pengguna menjadi data yang dapat ditindak lanjuti. Contoh: Quiz-Bot pada gambar di atas.
2. **Intents** : Ini dikonfigurasi oleh pengembang yang menunjukkan apa tujuan pengguna mungkin ketika dia membuat permintaan tertentu. Misalnya: pemesanan tiket pesawat atau memberikan feedback.

3. **Entities** : Membantu ekstrak informasi dari ucapan pengguna dengan bantuan prompt. Misalnya: "Pesan penerbangan" mungkin perlu seperti: kemana dan dari kota mana, tanggal, kelas dll sebagai entitas yang dicoba agen untuk mengekstrak dari pengguna melalui percakapan. Informasi yang diterima di sini dikirim untuk pemenuhan.
4. **Fulfilment** : Bagian dari percakapan ini memungkinkan Anda meneruskan permintaan dari bot Anda ke sumber eksternal dan mendapatkan respons dan mengirimkannya kembali kepada pengguna. Ini dicapai melalui Webhook. Menyiapkan webhook memungkinkan Anda meneruskan informasi dari intent yang cocok ke layanan web dan mendapatkan hasilnya.
5. **Entities** : Membantu ekstrak informasi dari ucapan pengguna dengan bantuan prompt. Misalnya: "Pesan penerbangan" mungkin perlu seperti: kemana dan dari kota mana, tanggal, kelas dll sebagai entitas yang dicoba agen untuk mengekstrak dari pengguna melalui percakapan. Informasi yang diterima di sini dikirim untuk pemenuhan.
6. **Integrations** : Dialogflow dapat diintegrasikan dengan platform lain seperti facebook Messenger, Slack, Line, Telegram, dll.
7. **Prebuilt Agents** : Pada bagian ini adalah tahap dimana pra produksi Agents anda untuk benar – benar rilis. Disini anda akan menentukan jenis Agent anda misalnya Agent untuk Alarm, Banking, dll
8. **Smalltalk** : Pada bagian ini anda dapat membuat Bot yang ramah dan cerewet tanpa coding dari Dialogflow.

Secara garis besar alur percakapan pada dialogflow adalah sebagai berikut.



(Gambar 1.4 alur percakapan dialogflow)

## Setting up Dialogflow account

Ok, bagian materi yang “membosankan” diatas sudah selesai. Saat nya kita mulai mengatur environment untuk kita membuat bot.

1. Silahkan kunjungi <https://dialogflow.com/>
2. Lalu pilih “Sign up for free”
3. Setelah itu Sign in dengan akun google anda.
4. Jika proses Sign in telah selesai maka akan menuju kehalaman <http://console.dialogflow.com/>.
5. Isi form seperti berikut :

Please review your account settings

Country or territory \*

Indonesia

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 \*

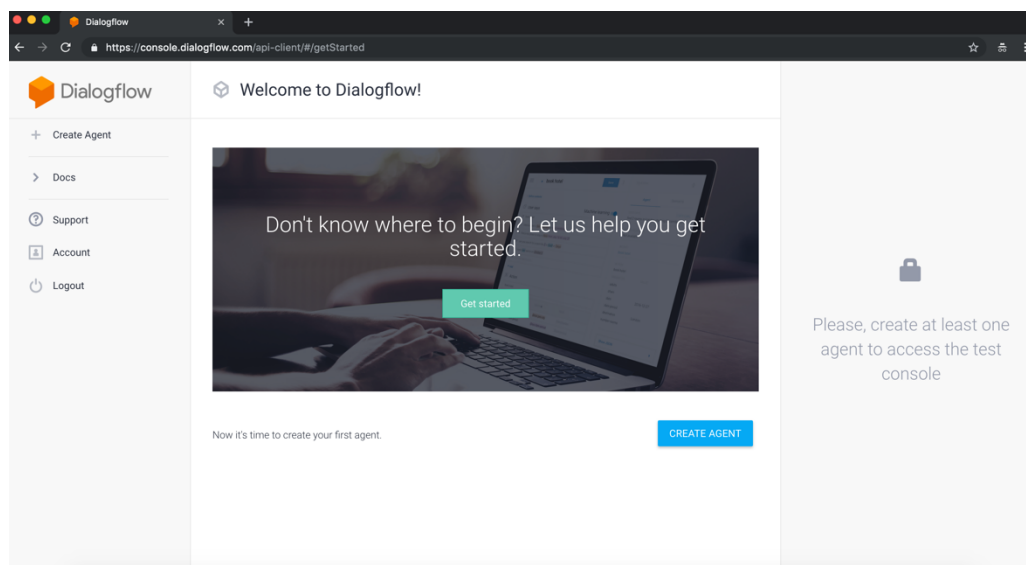
☒ Yes, I have read and accept the agreement.

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

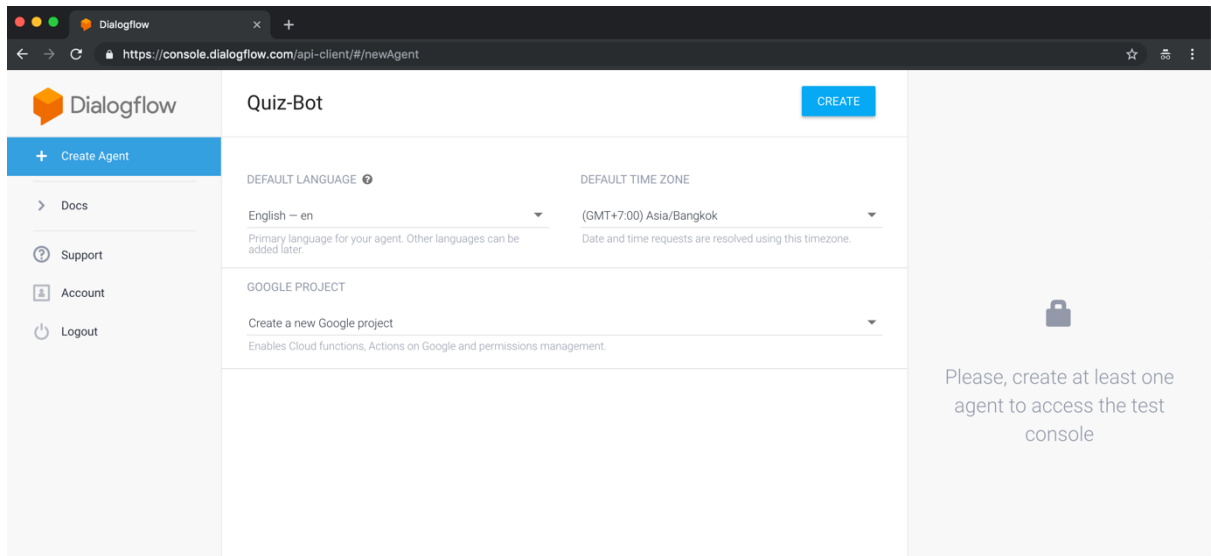
6. Jika semua selesai maka anda akan menemui halaman seperti ini



# Create new Agent

Nah dibagian ini kita akan mulai membuat bot, ikuti langkah – langkah dibawah dengan benar.

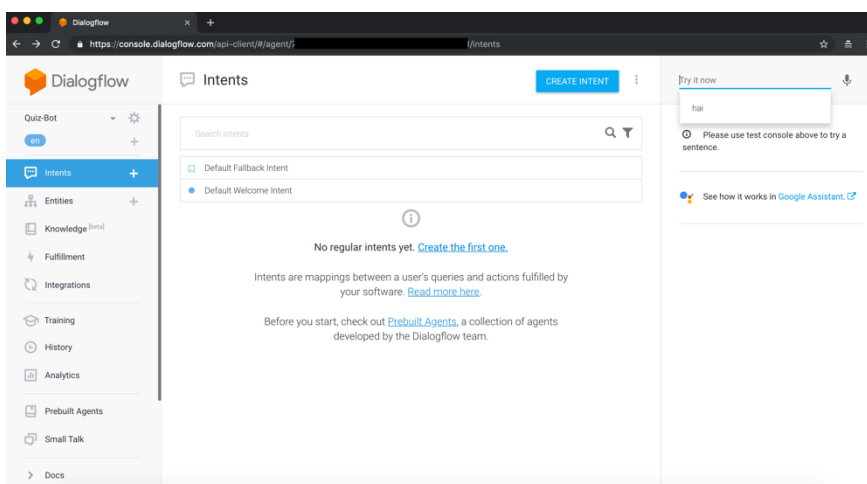
1. Create Agent dan isi form Agent seperti berikut.




2. Setelah itu pilih “Create”

Catatan : jika anda ingin integrasikan dengan Google Cloud bisa memilih “Google Project” anda. Namun pada modul ini kita tidak menggunakan Google Cloud

3. Jika sudah masuk kedalam halaman <http://console.dialogflow.com/> seperti ini.

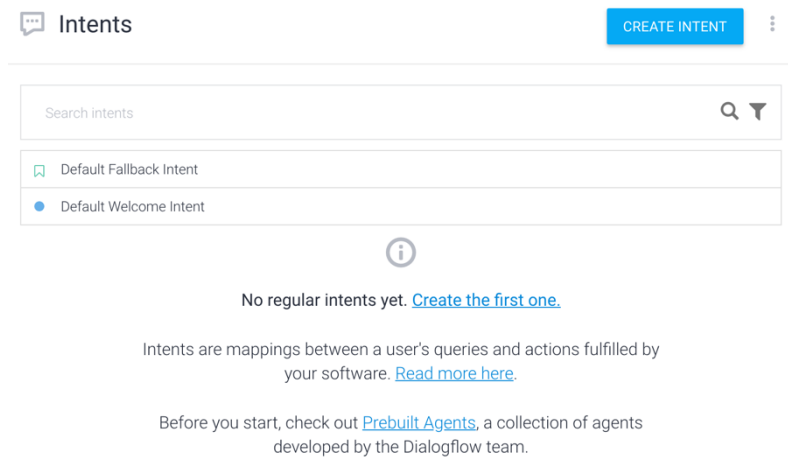


4. maka pilih tombol setting  untuk mendapatkan *API KEY*. Yang kita gunakan disini adalah “*Client access token*”.

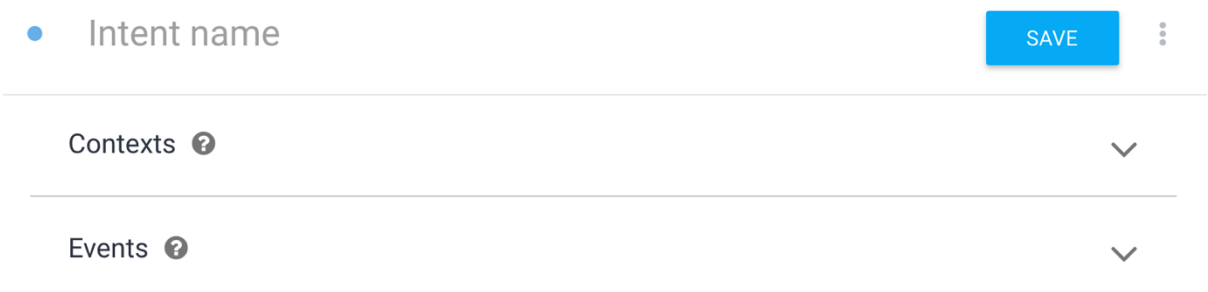
API KEYS (V1)

Client access token	<div></div>	 
Developer access token	<div></div>	

5. Kembali ke halaman Intents dan kita mulai membuat Intents baru.
6. Pada halaman Intents kita akan menemui default Intents dari dialogflow seperti berikut.



7. Mari kita buat Intents baru dengan menekan tombol “CREATE INTENT”
8. Setelah kita tekan tombol “CREATE INTENT” maka akan menuju halaman seperti berikut.






9. Kita isikan Intent Name dengan “quiz-1” lalu SAVE
10. Berikutnya kita isikan inputan user pada “Training phrases”. Contohnya seperti ini.






11. Jangan lupa setiap kali selesai melakukan input pada Training phrases tekan “Enter” pada windows atau “Return” pada Mac OS. Jika sudah menekan enter maka tampilan akan seperti ini.

Training phrases 


Search training phrases  


” siapa presiden pertama indonesia?

” siapa presiden pertama indonesia?

PARAMETER NAME	ENTITY	RESOLVED VALUE
geo-country	@sys.geo-country	indonesia 

12. Setelah membuat inputan user maka kita akan membuat respon untuk inputan user tersebut pada menu “Responses”

Responses 




**Execute and respond to the user**


Respond to your users with a simple message, or build custom rich messages for the integrations you support. [Learn more](#)



[ADD RESPONSE](#)

13. Tekan “ADD RESPONSE” untuk membuat respon baru

14. Buat Respon seperti berikut ini.


Responses 

[DEFAULT](#) 

Text response  

1	Ir. Soekarno
2	Enter a text response variant

[ADD RESPONSES](#)

☐ Set this intent as end of conversation 

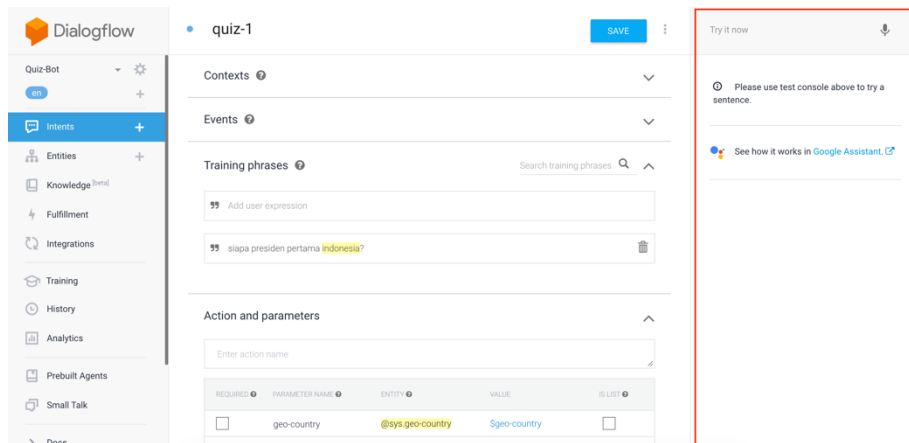
15. Setelah itu jangan lupa tekan tombol “SAVE”

 quiz-1

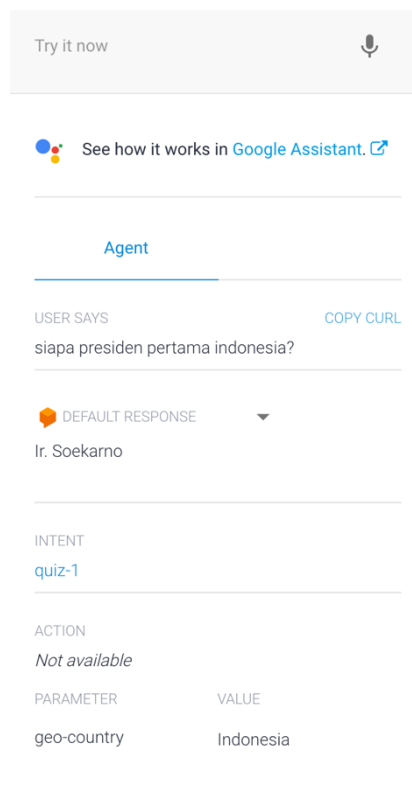
[SAVE](#) 

16. Mari kita coba apakah intent kita berjalan dengan sempurna

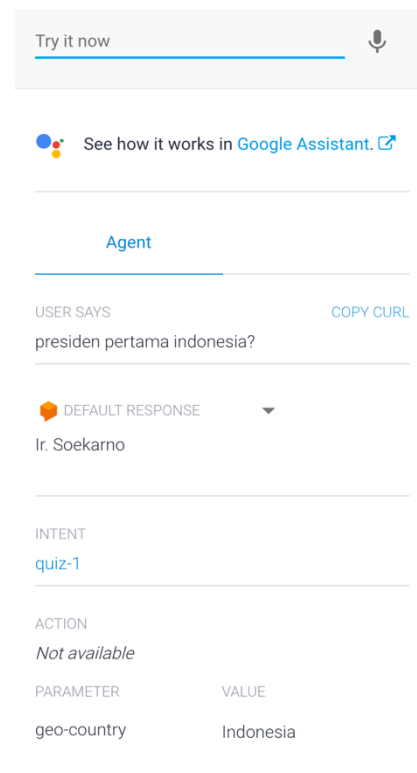
17. Caranya cukup mudah sekali yaitu dengan menggunakan simulasi yang ada disebelah kanan.



18. Mari kita coba. Ketikan “siapa presiden pertama indonesia?” maka respon Agent kita adalah “Ir.Soekarno”. hebatnya lagi adalah meskipun user inputkan kata yang tidak sesuai dengan keyword kita, Dialogflow akan tetap paham dan otomatis akan melakukan proses pendekatan kata secara natural karena dialogflow menggunakan Natural Language Understanding (NLU).



(Gambar 1.5 Sesuai keyword)



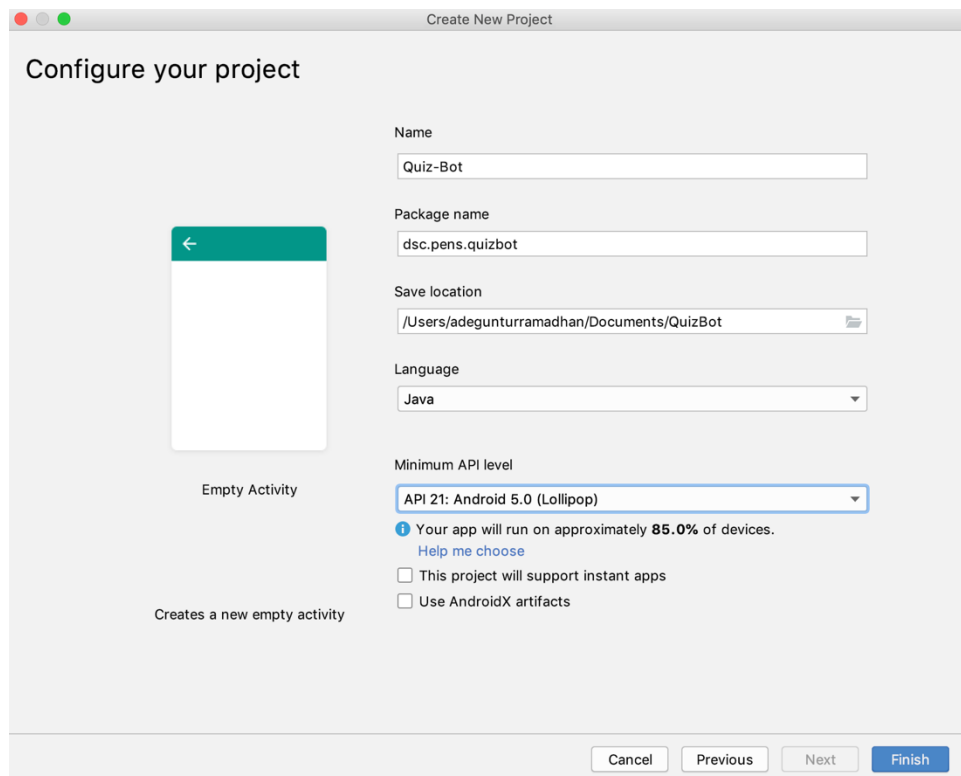
(Gambar 1.6 Tidak Sesuai keyword)

19. Baiklah kita telah selesai membuat Agent dan Intents. Berikutnya adalah kita membuat project android baru di android studio.

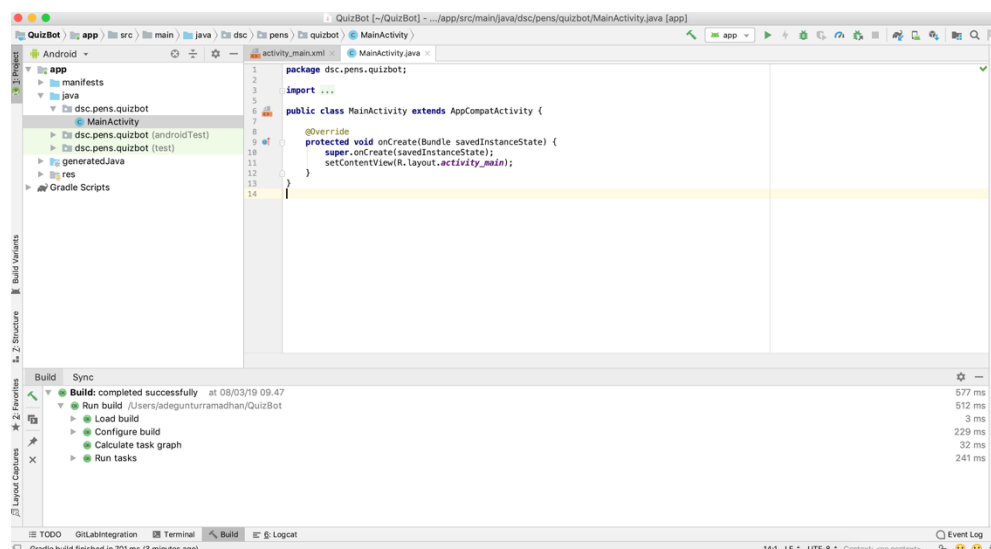
# Create New Project Android

Materi berikutnya adalah kita akan membuat aplikasi android dengan dialogflow yang telah kita buat dimateri sebelumnya. Ikuti dan perhatikan langkah - langkah dibawah ini.

1. Langkah pertama pastikan pada laptop/komputer anda terinstall Android Studio dan terkoneksi dengan internet.
2. Buka Android Studio anda dan buat project baru. Pada modul ini kita menggunakan java sebagai Bahasa pemrogramannya. Simpan project kedalam lokasi yang anda tentukan



3. Jika sudah tekan “Finisih” dan anda akan masuk kedalam project android studio seperti berikut ini



4. Buka build.gradle (Module: app) dan tambahkan beberapa library pada dependencies seperti berikut ini.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'

    // Dialogflow SDK dependencies
    implementation 'ai.api.sdk:2.0.7@aar'
    implementation 'ai.api:libai:1.6.12'
}
```

5. Setelah itu tambahkan buildConfigField pada defaultConfig seperti berikut ini. Jangan lupa paste kan client acces token yang sudah didapatkan sebelumnya.

```
defaultConfig {
    applicationId "dsc.pens.quizbot"
    minSdkVersion 21
    targetSdkVersion 28
    versionCode 1
    versionName "1.0"
    testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    buildConfigField "String", "ClientAccessToken", "\"letakan client acces token disini\""
}
```

6. Berikutnya buka manifests dan tambahkan uses-permission internet seperti berikut ini. Ingat tambahkan ini diluar tag application

```
<uses-permission android:name="android.permission.INTERNET" />
```

7. Buatlah file xml baru dengan cara click kanan pada res lalu New -> Drawable resource file beri nama bot\_bg\_bubble.xml setelah itu tambahkan kode seperti berikut ini.

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="@color/colorPrimary" />
    <corners android:topLeftRadius="1dp" android:radius="16dp" />
</shape>
```

8. Buatlah file xml baru dengan cara click kanan pada res lalu New -> Drawable resource file beri nama chatbot\_send\_btn.xml setelah itu tambahkan kode seperti berikut ini.

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="36dp"
    android:height="36dp"
    android:autoMirrored="true"
    android:viewportHeight="24.0"
    android:viewportWidth="24.0">
    <path
        android:fillColor="@color/colorAccent"
        android:pathData="M2.01,21L23,12 2.01,3 2,10L15,2 -15,2z" />
</vector>
```

9. Buatlah file xml baru dengan cara click kanan pada drawable lalu New -> Drawable resource file beri nama user\_bg\_bubble.xml setelah itu tambahkan kode seperti berikut ini.

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="@color/colorAccent" />
    <corners android:topRightRadius="1dp" android:radius="16dp" />
</shape>
```

10. Berikutnya buka values -> colors.xml dan sesuaikan colors.xml seperti berikut ini.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#0091EA</color>
    <color name="colorPrimaryDark">#01579B</color>
    <color name="colorAccent">#E65100</color>
</resources>
```

11. Berikutnya buka values -> styles.xml dan sesuaikan styles.xml seperti berikut ini.

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
        <item name="android:textColor">@android:color/white</item>
    </style>

</resources>
```

12. Buatlah layout baru dengan cara click kanan pada package layout lalu New -> Layout resource file beri nama bot\_msg\_layout.xml setelah itu tambahkan kode seperti berikut ini.

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="8dp">

    <LinearLayout
        android:id="@+id/botMsgLayout"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="start|center_vertical"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
        android:layout_marginEnd="16dp"
        android:layout_marginStart="8dp"
        android:background="@drawable/bot_bg_bubble"
        android:gravity="start|center_vertical"
        android:orientation="vertical"
        android:layout_marginRight="16dp"
        android:layout_marginLeft="8dp">

        <TextView
            android:id="@+id/chatMsg"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="4dp"
            android:padding="12dp"
            android:text="abcdefgh"
            android:textSize="18sp" />

        <FrameLayout
            android:id="@+id/container"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp">

    </FrameLayout>

</LinearLayout>
</FrameLayout>
```

13. Buatlah layout baru dengan cara click kanan pada package layout lalu New -> Layout resource file beri nama user\_msg\_layout.xml setelah itu tambahkan kode seperti berikut ini.

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="8dp">

    <LinearLayout
        android:id="@+id/userMsgLayout"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="end|center_vertical"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
        android:layout_marginEnd="16dp"
        android:layout_marginStart="8dp"
        android:background="@drawable/user_bg_bubble"
        android:gravity="end|center_vertical"
        android:orientation="vertical"
        android:layout_marginRight="16dp"
        android:layout_marginLeft="8dp">

        <TextView
            android:id="@+id/chatMsg"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="4dp"
            android:padding="12dp"
            android:text="abcdefgh"
            android:textSize="18sp" />

        <FrameLayout
            android:id="@+id/container"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp">

    </FrameLayout>

</LinearLayout>
</FrameLayout>
```

14. Buatlah class baru beri nama RequestTask. Jika error biarkan saja karena kita akan tambahkan kode di MainActivity.

```
package dsc.pens.quizbot;

import android.app.Activity;
import android.os.AsyncTask;

import ai.api.AIServiceContext;
import ai.api.AIServiceException;
import ai.api.android.AIDataService;
import ai.api.model.AIRequest;
import ai.api.model.AIResponse;

public class RequestTask extends AsyncTask<AIRequest, Void, AIResponse> {

    Activity activity;
    private AIDataService aiDataService;
    private AIServiceContext customAIServiceContext;

    RequestTask(Activity activity, AIDataService aiDataService, AIServiceContext customAIServiceContext){
        this.activity = activity;
        this.aiDataService = aiDataService;
        this.customAIServiceContext = customAIServiceContext;
    }

    @Override
    protected AIResponse doInBackground(AIRequest... aiRequests) {
        final AIRequest request = aiRequests[0];
        try {
            return aiDataService.request(request, customAIServiceContext);
        } catch (AIServiceException e) {
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(AIResponse aiResponse) {
        ((MainActivity)activity).callback(aiResponse);
    }
}
```

15. Berikutnya tambahkan kode pada activity\_main.xml seperti berikut ini.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <RelativeLayout
        android:id="@+id/inputLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:background="@android:color/white"
        android:gravity="bottom"
        android:paddingStart="8dp"
        android:paddingTop="8dp"
        android:paddingEnd="8dp"
        android:paddingBottom="9dp">

        <ImageView
            android:id="@+id/sendBtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_alignParentRight="true"
            android:paddingTop="4dp"
            android:src="@drawable/chatbot_send_btn" />

        <EditText
            android:id="@+id/queryEditText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentStart="true"
            android:layout_alignParentLeft="true"
            android:layout_toStartOf="@+id/sendBtn"
            android:layout_toLeftOf="@+id/sendBtn"
            android:imeOptions="actionSend"
            android:inputType="text"
            android:paddingTop="4dp"
            android:textSize="18sp" />

    </RelativeLayout>

    <ScrollView
        android:id="@+id/chatScrollView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@+id/inputLayout">

        <LinearLayout
            android:id="@+id/chatLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical" />

    </ScrollView>
</RelativeLayout>
```

16. Tambahkan compileOptions pada build.gradle (Module: app) didalam Android { } seperti berikut ini.

```
compileOptions {
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}
```

17. Terakhir tambahkan kode berikut ini pada MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    private static final String TAG = MainActivity.class.getSimpleName();
    private static final int USER = 10001;
    private static final int BOT = 10002;

    private String uuid = UUID.randomUUID().toString();
    private LinearLayout chatLayout;
    private EditText queryEditText;

    // Android client
    private AIRequest aiRequest;
    private AIDataService aiDataService;
    private AIServiceContext customAIServiceContext;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        final ScrollView scrollView = findViewById(R.id.chatScrollView);
        scrollView.post(() -> scrollView.fullScroll(ScrollView.FOCUS_DOWN));

        chatLayout = findViewById(R.id.chatLayout);

        ImageView sendBtn = findViewById(R.id.sendBtn);
        sendBtn.setOnClickListener(this::sendMessage);

        queryEditText = findViewById(R.id.queryEditText);
        queryEditText.setOnKeyListener((view, keyCode, event) -> {
            if (event.getAction() == KeyEvent.ACTION_DOWN) {
                switch (keyCode) {
                    case KeyEvent.KEYCODE_DPAD_CENTER:
                    case KeyEvent.KEYCODE_ENTER:
                        sendMessage(sendBtn);
                        return true;
                    default:
                        break;
                }
            }
            return false;
        });
        initChatbot();
    }

    private void initChatbot() {
        final AIConfiguration config = new AIConfiguration(BuildConfig.ClientAccessToken,
            AIConfiguration.SupportedLanguages.English,
            AIConfiguration.RecognitionEngine.System);
        aiDataService = new AIDataService(context: this, config);
        customAIServiceContext = AIServiceContextBuilder.buildFromSessionId(uuid); // helps to create new session whenever app restarts
        aiRequest = new AIRequest();
    }

    private void sendMessage(View view) {
        String msg = queryEditText.getText().toString();
        if (msg.trim().isEmpty()) {
            Toast.makeText(context: MainActivity.this, text: "Please enter your query!", Toast.LENGTH_LONG).show();
        } else {
            showTextView(msg, USER);
            queryEditText.setText("");
            aiRequest.setQuery(msg);
            RequestTask requestTask = new RequestTask(activity: MainActivity.this, aiDataService, customAIServiceContext);
            requestTask.execute(aiRequest);
        }
    }

    public void callback(AIResponse aiResponse) {
        if (aiResponse != null) {
            // process aiResponse here
            String botReply = aiResponse.getResult().getFulfillment().getSpeech();
            Log.d(TAG, msg: "Bot Reply: " + botReply);
            showTextView(botReply, BOT);
        } else {
            Log.d(TAG, msg: "Bot Reply: Null");
            showTextView(message: "There was some communication issue. Please Try again!", BOT);
        }
    }

    private void showTextView(String message, int type) {
        FrameLayout layout;
        switch (type) {
            case USER:
                layout = getUserLayout();
                break;
        }
    }
}
```




```

        case BOT:
            layout = getBotLayout();
            break;
        default:
            layout = getBotLayout();
            break;
    }
    layout.setFocusableInTouchMode(true);
    chatLayout.addView(layout); // move focus to text view to automatically make it scroll up if softfocus
    TextView tv = layout.findViewById(R.id.chatMsg);
    tv.setText(message);
    layout.requestFocus();
    queryEditText.requestFocus(); // change focus back to edit text to continue typing
}
}

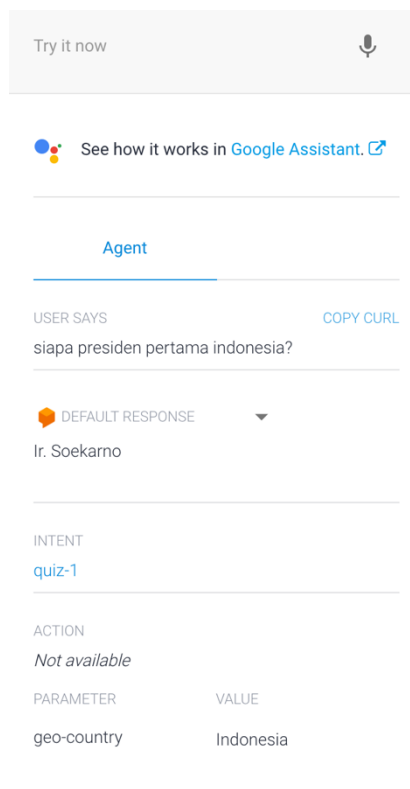
FrameLayout getUserLayout() {
    LayoutInflater inflater = LayoutInflater.from(MainActivity.this);
    return (FrameLayout) inflater.inflate(R.layout.user_msg_layout, root: null);
}

FrameLayout getBotLayout() {
    LayoutInflater inflater = LayoutInflater.from(MainActivity.this);
    return (FrameLayout) inflater.inflate(R.layout.bot_msg_layout, root: null);
}
}
}

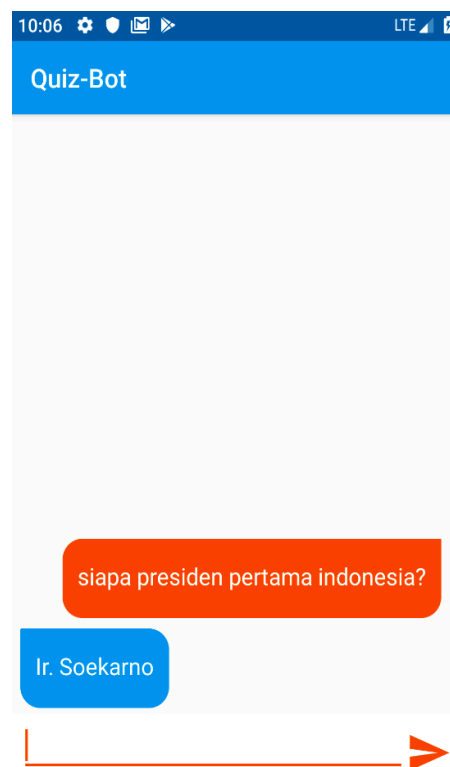
```

18. Mari kita coba run project tersebut 

19. Hasil respon bot sesuai dengan Intents yang kita buat tadi diawal.



(Gambar 1.7 Test Intents)



(Gambar 1.8 Aplikasi Quiz-Bot)

Terimakasih telah mengikuti modul yang pertama ini sampai selesai. Jangan lupa untuk mengikuti modul berikutnya pada event selanjutnya karena modul ini terbagi menjadi 3 bagian.