

CHAPTER 4

ANALYSIS AND DESIGN

4.1 Analysis

This contact application works so that users can still have contacts that are synchronized even if they change their cellphone or lose their cellphone. To make this application an API is needed. The API itself is a link between applications. How to get the API is by using OAuth.02 from Google. Google provides a free platform to retrieve the API according to their individual needs. The first step is to have to create a project on the firebase for the purpose of signing in using a Google account, then take the credentials from the Google API Contacts which will be entered into the gradle on the Android studio. The application that is created requires an access in order to log in using firebase. Actually, if you directly use google fire, you can, but when you make the first time we will not find an authentication in it. Therefore when creating the login activity, the first time that is done to get the client ID is to use authentication from the firebase. After that what has been created in the firebase will all be read and automatically saved on the Google Fire console. Contacts API is created through the Google console by creating a project in it using the Android Studio project package along with SHA1. Only can create one project with the same SHA1, if different Google console will immediately reject.

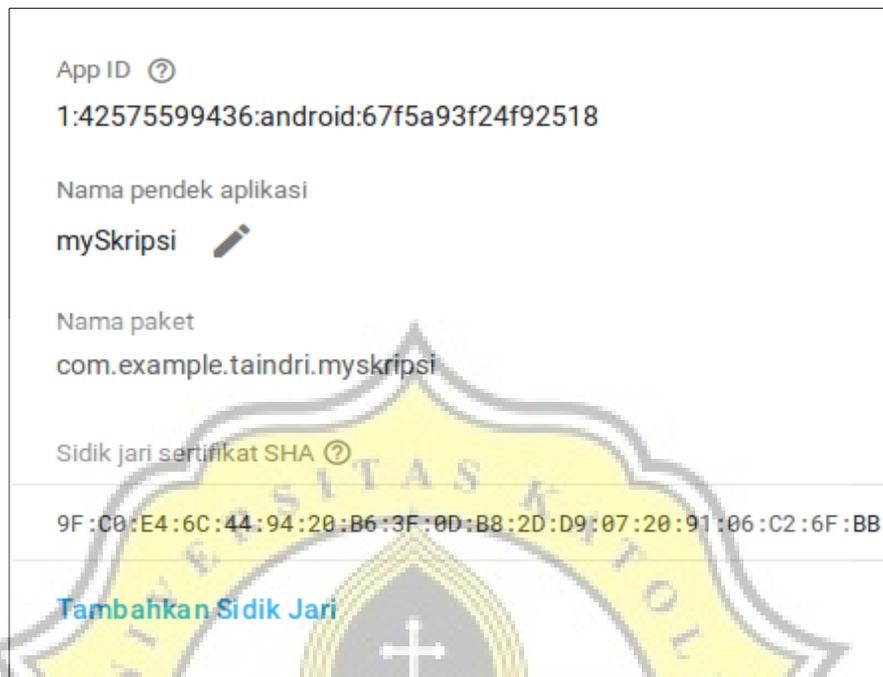


Illustration 4.1 : SHA1

In the short name, the application is filled in with the name of the application created in Android Studio. The package name is filled according to when creating a project on Android Studio, each class will appear with the same package name. Then the last one to enter fingerprint or SHA1 which can be seen in the android studio project is in the Sign In Report.

ID klien OAuth 2.0			
<input type="checkbox"/> Nama	Tanggal dibuat ▾	Jenis	ID Klien
<input type="checkbox"/> Android client for taindri.example.com.skripsiku (auto created by Google Service)	Okt 11, 2018	Android	850287614524-9oq9v13ci09caghdea20pj26n5o7jptd.apps.googleusercontent.com ✎ 🗑

Illustration 4.2 : Kredensial

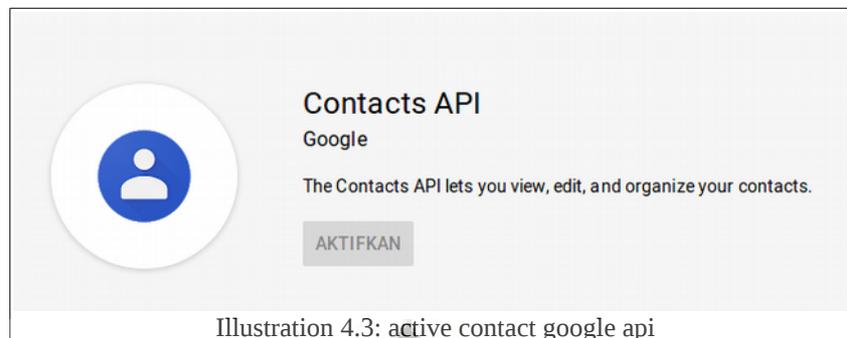


Illustration 4.3: active contact google api

After SHA1 has been added successfully, the next step is to activate the API contact on google. When activated, the project that will automatically be created in the firebase will appear and has activated the API contact and can immediately know its credentials or JSON format files as shown in illustration 4.3. In making the application do not use a username and password. The thing needed to log in is access to permissions from Google account itself. The application issued by Google Inc. also does not use usermane or password, which is directly logged in using a google account. To display contacts that have been synchronized through the mobile phone, an API is needed which functions as a link between the application or the web. What we get when registering a project to have the API itself is a JSON format file. JSON is a format for sharing data. As we can see from its name, JSON is derived from the Java programming language, but this format is available for many other languages including Python, Ruby, PHP, and Java. JSON is usually pronounced like the name "Jason." Generally JSON is often used in web applications to send data from the server to the browser.

The JSON format file is imported into the Android Studio project. After that it is necessary to add a library so that the API can function in the application that will be created. The library entered into the gradle project must adjust the version of the Android studio SDK used. Then to display the list of contacts used is the RecyclerView layout. The function is to be able to display and scroll easily regardless of the contact because it will be looped later according to the number of contacts displayed. Using RecyclerView is easier because the contacts you want to

display are easier, simpler, and can scroll properly. Continue to create classes to display features in contacts. An application that is created contains the add contact feature, edit contacts, and delete contacts. In the display section when the application displays the contact list, this feature will pop up when we click on a contact. The pop up dialog contains View Contacts, Update Contacts, and Delete Contacts. Made like that so that there are not many buttons and look more minimalist, so users who try will not be confused.



Illustration 4.4: oauth playground

In order for API contacts to work, it is necessary to use the scope provided by Google Inc. Scope is used to define what information will be requested. This application will ask for all information such as name, phone and birthday data. This scope can be seen in Oauth Playground. Each API that requires the scope will vary.

4.2 Design

The flowchart design to make this project on the Android platform is as shown under :

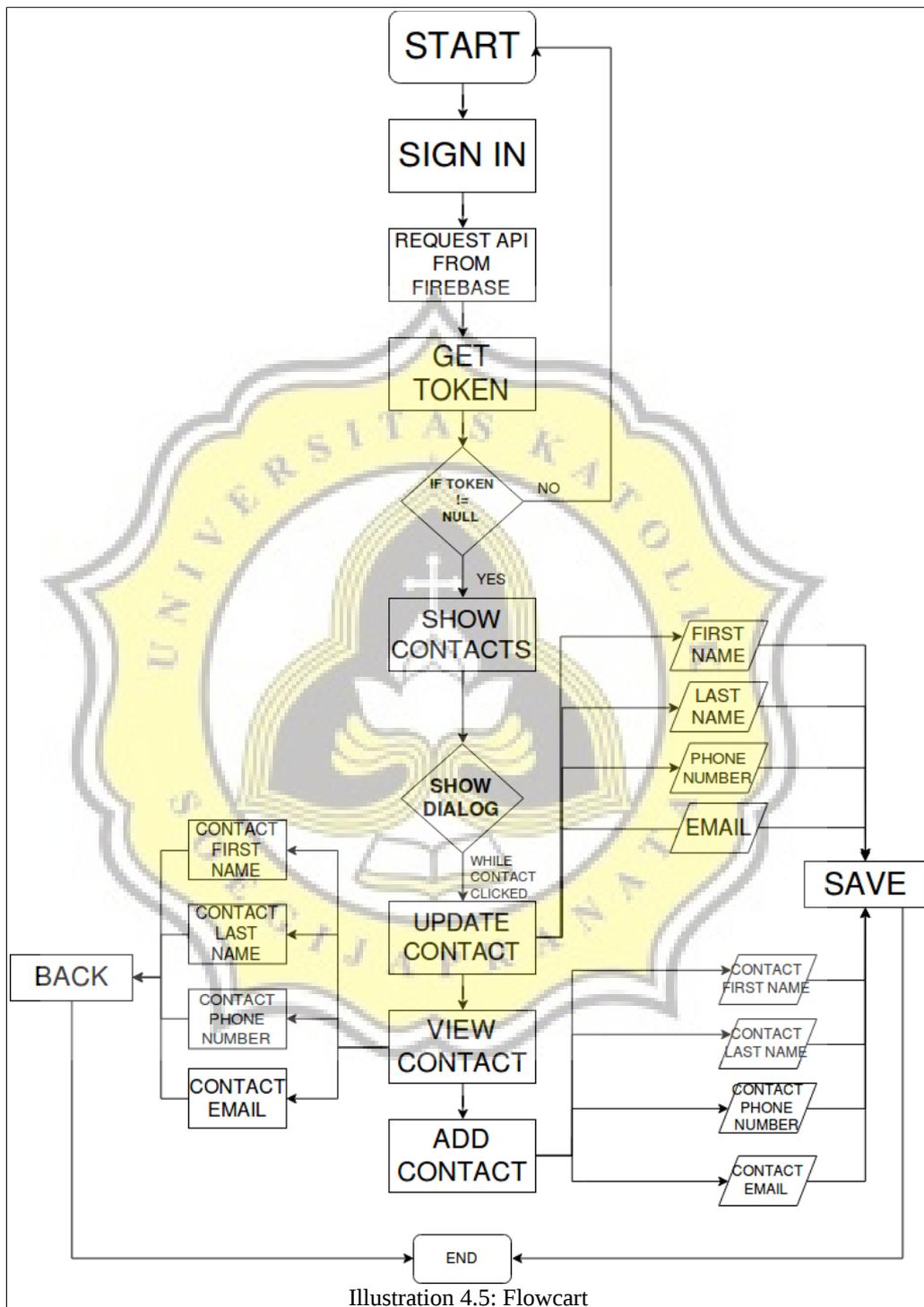


Illustration 4.5: Flowcart

Based on the picture above is how the API works directly on the application in the android studio. When Signing in using a google account is required for API requests from firebase. Because for login authentication must go through the firebase first. After getting the token, it can enter the application, if it fails it will be directed to the login activity. The main cause of failure is usually due to an inappropriate password and email, or internet connection. Next the user can see the contact list that has been synchronized to the google account either phone number or email. In the contact list, the user can click and a dialog will appear that will fill the view of contacts, update contacts, and delete contacts. When you want to update automatically it will be saved, if you will not do any activity you can click the back button which means the process is complete.

