

Android Based Application of TiPS to Teach Verb Conjugations in Simple Statements in the English Language

Febe Maedjaja*¹, Yosua Susanto²

^{1,2}Universitas Kristen Immanuel; Jl. Solo Km. 11,1, Kalasan, DIY, (0274)496256
e-mail: *febe@ukrimuniversity.ac.id, yosua.h.susanto@gmail.com

Abstract

Wide spread confusions about the verb conjugations in English and the phenomena of open source Android and its applications have led us to this study. We studied the lesson materials, we designed an Android based application, we did test runs of the application, and we surveyed the responses of sample users. The study yielded an application written in Java, developed on Eclipse IDE, and using SQLite as its database management system that runs on Android platform. The test runs of the application on five types of smartphone devices with different Android versions showed successful evaluations over seven criteria. The responses of thirty sample users showed an overwhelming 98.34% positive responses over ten criteria of the application, which led us to safely conclude that the application successfully accepts our alternate hypothesis, that . Survey analysis led us to the successful acceptance of our alternative hypothesis that an Android based TiPS lessons application that focuses on verb conjugations helps Indonesian students in learning English grammar.

Keywords—Verb conjugation, Simple statement, TiPS teaching method, Android, SQLite

1. INTRODUCTION

Three significant phenomena are happening around us that led to this study. “I am sit.” “Why are your child crying?” “Do you can come to my house tomorrow?” “Do we going to the campus?” “He is plays badminton every evening.” These are just some examples of common mistakes we often hear from Indonesians who are studying English. The problem is obvious. English verb conjugations are very confusing, mainly because there is no concept of tenses in the Indonesian language. We have also observed another phenomenon of the dominance as well as the upward trend of the use of Android operating system among Indonesian smartphone users ^[1]. The third phenomenon is the rise of open source software ^[2]. The popularity of Android is just one proof, not to mention other open source applications that run on it.

The purpose of this study is to produce a very accessible tool that focuses on strengthening the knowledge and habit of using the correct verb for every simple statement in English.

In this study we set out to first find other literatures on teaching verb conjugations, although our lessons will be based on the TiPS method. We will also study open source Android and Android based applications and use them to develop the application. Our product will be tested on several devices and sample users.

Our alternative hypothesis is, that an Android based TiPS lessons application that focuses on verb conjugations helps Indonesian students in learning English grammar.

We assume that the students who will participate in the validation of this study are Indonesians who are somewhat familiar with using a smartphone and have had a minimum of high school diploma.

A few restrictions are applied in this research:

1. This is not a linguistic study, and we have no claim of linguistic expertise. We look at the problem of understanding English grammar from the point of view of a scientist and an IT person who had to learn English as a second language.
2. The application will be Android based.

2. RESEARCH DESIGN

The three main areas that must be examined in this study are the design of the lesson materials, the design of the application, the testing of the application on several devices, and surveying the responses of sample users.

The lessons will focus on verb conjugations. Because of this, the exercises must be designed such that in expressing a simple statement in English, all the other components as subject, object, etc have been ordered according to the proper formula. In other words, steps 1 and 2 of TiPS are being done by the application. The user can focus in step 3, i.e. to find the correct verb conjugation in relationship to the tense and subject. The language of instruction will be Indonesian, since understanding is the goal of the lessons.

For the sake of simplicity yet comprehensiveness, the lessons will cover fifteen tenses: Descriptive, Continuous, Simple, Perfect, and Perfect Continuous in the Present, Past and Future tenses^[3,4,5]. We need to extend the formulas available in the past publication on TiPS to include all of these fifteen tenses. These additional formulas can be seen in Tables 1 through 7 are the complementary formulas for Present Perfect Continuous, Past Perfect Continuous, Future Descriptive, Future Simple, Future Continuous, Future Perfect, and Future Perfect Continuous respectively.

Table 1 Present Perfect Continuous Tense with action verbs

Pattern	Formulas
+	S <u>have/has been V_{ing}</u> (O how where when). V
-	S <u>have/has not been V_{ing}</u> (O how where when). V
?Y/N	<u>Have/Has</u> S been <u>V_{ing}</u> (O how where when)? auxV mainV
?Expl	QW <u>have/has</u> S been <u>V_{ing}</u> (O how where when)? auxV mainV
?S	QWS <u>have/has been V_{ing}</u> (O how where when)? V

Table 2 Past Perfect Continuous Tense with action verbs

Pattern	Formulas
+	S <u>had been V_{ing}</u> (O how where when). V
-	S <u>had not been V_{ing}</u> (O how where when). V
?Y/N	<u>Had</u> S been <u>V_{ing}</u> (O how where when)? auxV mainV
?Expl	QW <u>had</u> S been <u>V_{ing}</u> (O how where when)? auxV mainV
?S	QWS <u>had been V_{ing}</u> (O how where when)? V

Table 3 Future Descriptive with the verb "be"

Pattern	Formulas
+	S <u>will/shall be</u> (C where when). V
-	S <u>will/shall not be</u> (C where when). V
?Y/N	<u>Will/Shall</u> S <u>be</u> (C where when)? auxV mainV
?Expl	QW <u>will/shall</u> S <u>be</u> (C where when)? auxV mainV
?S	QWS <u>will/shall be</u> (C where when)? V

Table 4 Future Simple tense with action verbs

Pattern	Formulas
+	S <u>will/shall V₁</u> (O how where when). V
-	S <u>will/shall not V₁</u> (O how where when). V
?Y/N	<u>Will/Shall</u> S <u>V₁</u> (O how where when)? auxV mainV
?Expl	QW <u>will/shall</u> S <u>V₁</u> (O how where when)? auxV mainV
?S	QWS <u>will/shall V₁</u> (O how where when)? V

Table 5 Future Continuous tense with action verbs

Pattern	Formulas
+	S <u>will/shall be V₁-ing</u> (O how where when). V
-	S <u>will/shall not be V₁-ing</u> (O how where when). V
?Y/N	<u>Will/Shall</u> S <u>be V₁-ing</u> (O how where when)? auxV mainV
?Expl	QW <u>will/shall</u> S <u>be V₁-ing</u> (O how where when)? auxV mainV
?S	QWS <u>will/shall be V₁-ing</u> (O how where when)? V

Table 3.6
Future Perfect tense with action verbs

Pattern	Formulas
+	S <u>will/shall have V₃</u> (O how where when). V
-	S <u>will/shall not have V₃</u> (O how where when). V
?Y/N	<u>Will/Shall</u> S <u>have V₃</u> (O how where when)? auxV mainV
?Expl	QW <u>will/shall</u> S <u>have V₃</u> O how where when)? auxV mainV
?S	QWS <u>will/shall have V₃</u> (O how where when)? V

Table 7 Future Perfect Continuous Tense with action verbs

Pattern	Formulas
+	S <u>will/shall have been V₁ing</u> (O how where when). V

-	S <u>will/shall not have been V_{ing}</u> (O how where when) V
?Y/N	<u>Will/Shall</u> S <u>have been V_{ing}</u> (O how where when)? auxV mainV
?Expl	QW <u>will/shall</u> S <u>have been V_{ing}</u> (O how where when)? auxV mainV
?S	QWS <u>will/shall have been V_{ing}</u> (O how where when)? V

The lessons will focus on verb conjugations. Because of this, the exercises must be designed such that in expressing a simple statement in English, all the other components as subject, object, etc have been ordered according to the proper formula. In other words, steps 1 and 2 of TiPS are being done by the application. The user can focus in step 3, i.e. to find the correct verb conjugation in relationship to the tense and subject. The language of instruction will be Indonesian, since understanding is the goal of the lessons.

The Use Case Diagram in Figure 1 shows the actors, use cases, and interaction between them in the application we are designing, while Table describes each component.

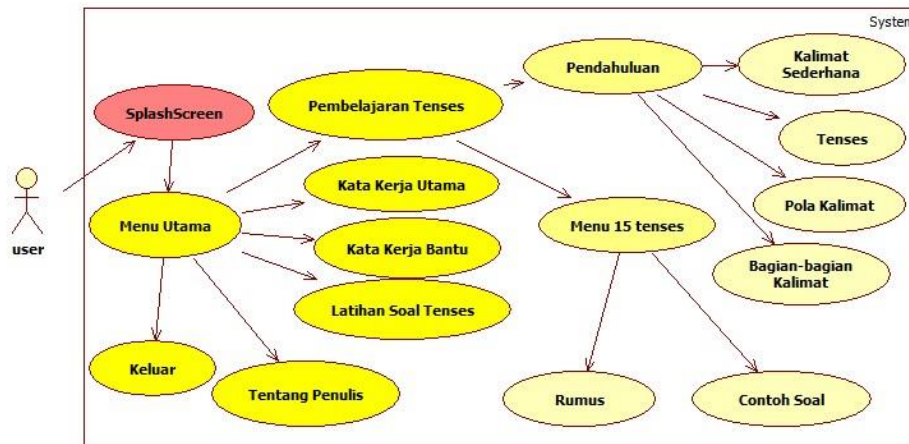


Figure 1 The Use Case Diagram

Table 8 The components of the Use Case Diagram

Use Cases (in Indonesian)	Use Cases (in English)	Descriptions
User	User	User is each individual that uses this application.
Splash Screen	Splash Screeb	Spalsh screen is the initial display when the application is first opened by the user.
<i>Menu Utama</i>	Main Menu	User can choose one of the menus in this main menu such as <i>pembelajaran tenses, kata kerja utama, kata kerja bantu, latihan soal, tentang penulis, keluar.</i>
<i>Pembelajaran Tenses</i>	Lessons on Tense	User can choose Pendahuluan or lessons on the fifteen <i>tenses</i> .
<i>Pendahuluan</i>	Introduction	User can choose one of the four submenus: <i>kalimat sederhana,</i>

		<i>tenses, pola kalimat, and bagian-bagian kalimat.</i>
<i>Kalimat Sederhana</i>	Simple Statements	User will have access to lessons on simple statements.
Tenses	Tenses	User will have access to lessons on tenses, past tenses, present tenses, dan future tenses.
<i>Pola Kalimat</i>	Sentence Patterns	User will have access to lessons on the various patterns of sentences.
<i>Bagian-bagian Kalimat</i>	Components of a Simple Sentence	User can see the components that make up a simple statement.
<i>Menu 15 tenses</i>	The fifteen tenses	User can choose a tense and learn the formula or examples of the tense.
<i>Rumus</i>	Formulas	User will have access to the formula of the tense that he/she has chosen in the previous menu.
<i>Contoh Soal</i>	Examples	User can see an example of the tense that he/she has chosen in the previous menu.
Kata Kerja Utama	Main Verb	User can learn the various conjugations of a verb and its meaning in Indonesian.
Kata Kerja Bantu	Auxiliary Verb	User can learn the various auxiliary verbs in English.
Latihan Soal <i>Tenses</i>	Tense Exercises	User can do exercises of a certain tense. In this menu fifteen tenses are available, and for each tense ten practice exercises will appear randomly. At the end of each exercise User will see his/her score.
Tentang Penulis	About the Author	User can see the data of the author.
Keluar	Exit	User exits the application.

The Entity Relationship Diagram of the database can be seen on Figure 2. The table for Names of Tenses (“Tenses”) can be seen on Table 9, Patterns of Tenses (“Pola Tenses”) on Table 10, Examples (“*Contoh Soal*”) on Table 11, Auxiliary Verbs (“*Kata Kerja Bantu*”) on Table 12, Exercises (“*Latihan Soal*”) on Table 13, and Main Verbs (“*Kata Kerja Utama*”) on Table 14.

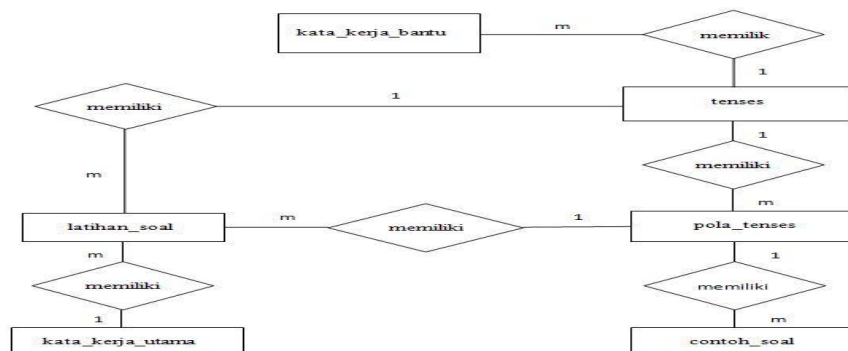


Figure 2 The Entity Relationship Diagram

Table 9 Table for Names of Tenses (“Tenses”)

Field	Type	Constraint
id	Integer	Primary key
nama_tenses	Varchar(50)	Not null
syarat_dan_ketentuan	Longtext	Not null

Table 10 Table for Patterns of Tenses (“Pola Tenses”)

Field	Type	Constraint
Id	Integer	Primary key
id_tenses	Integer	Not null, foreign key tenses(id)
nama_pola_tenses	Varchar(50)	Not null
Rumus	Varchar(50)	Not null

Table 11 Table for Examples (“Contoh Soal”)

Field	Type	Constraint
Id	Integer	Primary key
id_pola_tenses	Integer	Not null, foreign key pola_tenses(id)
soal_indonesia	Varchar(100)	Not null
soal_inggris	Varchar(100)	Not null

Table 12 Table for Auxiliary Verbs (“Kata Kerja Bantu”)

Field	Type	Constraint
Id	Integer	Primary key
id_tenses	Integer	Not null, foreign key tenses(id)
Aux_verb	Varchar(20)	Not null
Subjek	Varchar(50)	Not null

Table 13 Table for Exercises (“Latihan Soal”)

Field	Type	Constraint
Id	Integer	Primary key
Id_tenses	Integer	Not null, Foreign key tenses(id)
id_pola_tenses	Integer	Not null, Foreign key pola_tenses(id)
id_kata_kerja_utama	Integer	Not null, Foreign key kata_kerja_utama(id)
komponen_soal1	Varchar(50)	
komponen_soal2	Varchar(50)	
komponen_soal3	Varchar(50)	
komponen_soal4	Varchar(50)	
komponen_soal5	Varchar(50)	
jawaban_benar_kk_utama	Varchar(50)	
jawaban_benar_kk_bantu	Varchar(50)	
posisi_kku	Varchar(50)	
Posisi_kkb	Varcher(50)	
Arti	Varchar(150)	Not null

Table 14 Table for Main Verbs (“Kata Kerja Utama”)

Field	Type	Constraint
id	Integer	Primary key
v_1	Varchar(50)	Not null
v_2	Varchar(50)	Not null
v_3	Varchar(50)	Not null
v_ing	Varchar(50)	Not null
v1_+s/es	Varchar(50)	Not null
arti	Varchar(50)	Not null

The user interfaces are designed by grouping similar functions in one pulldown sub-menu. Figure 3 below shows the menu interface structure of the application.

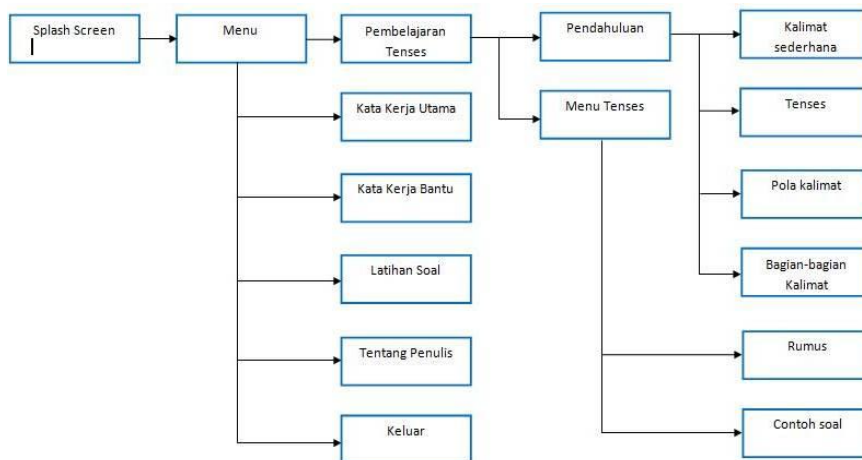


Figure 3 Design of the menu interface structure of the application

Figure 4 shows the design of the splash screen. The user may tap any place on the screen to move on to the next menu. All the menu screens on Figure 3 are designed, while the menus of each screen would follow the descriptions on Table 8. As an example, the scrollable screen for Main Verbs (*Kata Kerja Utama*) can be seen on Figure 5. The user can use the “Search” function by inputting the word and touching the “Cari” (Search) button. Figure 6 shows the design of the screen display of the Main Verb after a user search for the verb “eat.”



Figure 4 Design of the splash screen



Figure 5 Design for the scrollable Main Verb (*Kata Kerja Utama*) screen



Figure 6 Design of the Main Verb screen after a search for the verb "eat"

We would test the application on five smartphone devices and Android versions. The devices and their specifications can be seen on Table 15.

Table 15
Smartphone devices and Android versions for application testing

No.	Smartphone type	Android version	RAM	Resolution	Ex Display size
#1	Advans S5E	Version 4.2 (Jelly Bean)	512 MB	720x1280	5"
#2	Asus Zenfone 4	Version 4.4 (Kitkat)	1024 MB	480x800	4"
#3	Asus Zenfone 4	Version 5.0 (Lollipop)	1024 MB	480x800	4"
#4	Smartfren Andromax i	Version 4.0 (Ice Cream Sandwich)	512 MB	480x800	4"
#5	Evercross A12	Version 4.4 (Kitkat)	256 MB	320x480	3,5"

We designed a questionnaire in Indonesian to survey the responses of our sample. The English version of the questions in the questionnaire can be seen on Table 16. The responses will be categorized in five groups: Strongly Agree (SA), Agree (A), N (Neither agree nor disagree), Disagree (D), and SD (Strongly Disagree).

Table 16
Survey Questionnaire

No.	Question	SA	A	N	D	SD
P1	The lessons explain simple statements and the fifteen kinds of tenses in the English language clearly.					

P2	Exercises that focus on verb conjugations in simple statements as presented in the lessons help users in learning English tenses effectively and efficiently:					
	2.a. Users are assisted in choosing the correct tenses.					
	2.b. Users are assisted in choosing the correct verb conjugation that corresponds with a certain tense.					
	2.c. Users are assisted to understand, that sometimes a verb consists of an auxiliary verb and a main verb.					
	2.d. Users are assisted to understand the five widely used patterns of simple statements in the English language.					
	2.e. Users are assisted in making a correct simple statement in English.					
P3	The logical organization of menus and submenus makes it easy for the users to navigate through the application.					
P4	The display is attractive.					
P5	Access to the contents in each menu is fairly fast.					

3. RESULTS and DISCUSSIONS

This study yielded an application that focuses on step 3 of the TiPS method, i.e. it focuses on the verb conjugations. The application was developed in Java language by utilizing Eclipse IDE and SQLite as the database management system to be run on Android. The test runs of the application were then performed on several devices, and the application was tested on some volunteer samples.

The application contains lessons that cover fifteen tenses: Descriptive, Continuous, Simple, Perfect, and Perfect Continuous in the Present, Past and Future tenses. In the exercises the users are assisted in that all other components as subject, object, etc are presented and ordered according to the proper formula. The language of instruction is Indonesian. Users navigate through the lessons by using the menu interfaces.

The users are greeted by the splash screen as shown on Figure 7 while the program is being loaded. They then tap the screen anywhere to move on the the next menu.



Figure 7 The splash screen

All the designs in Chapter 3 have been developed. The user can scroll up and down the pages when there are more than one page worth of information.

As an example, the design of screen for Main Verbs (*Kata Kerja Utama*) on Figure 5 has been developed into Figure 8. The user can use the “Search” function by inputting the word and touching the “Cari” (Search) button. Figure 9 shows the edit display for a user searching for a certain main verb. Figure 10 shows the next display, while Figure 11 is the implementation of Figure 6 which shows the screen display of the main verbs that consist the word “eat.”



Figure 8 Scrollable Menu interface for Main Verb

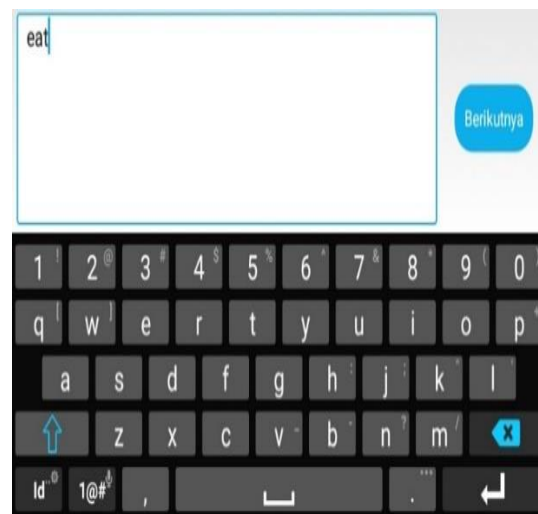


Figure 9 Edit interface for searching a main verb



Figure 10 Display after a search for “eat”



Figure 11 Display of a successful search for “eat”

SQLite was used to build the database, and SQLite DBHelper to connect the database with the main class activity in the program. The getter function is used to produce a value of a calculation, while the setter function is used to validate and to set data before it is input in a property. Each table in the database has to have a class getter and setter. The main activity displays and manage a user interface page as the area where a user interacts with the application. It also allows for the interaction between a few user interface

pages. The Android manifest contains all the property, class and other configurations in the development of the application. Examples of the script for these can be seen in Figures 12-15.

```
public SQLiteDatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
    ctx = context;
}

public void CopyDataBaseFromAsset() throws IOException {
    InputStream myInput = ctx.getAssets().open(DATABASE_NAME);
    String outFileName = getDatabasePath();
    File f = new File(ctx.getApplicationInfo().dataDir + DB_PATH_SUFFIX);
    if (!f.exists())
        f.mkdir();
    OutputStream myOutput = new FileOutputStream(outFileName);
    byte[] buffer = new byte[1024];
    int length;
    while ((length = myInput.read(buffer)) > 0) {
        myOutput.write(buffer, 0, length);
    }
    myOutput.flush();
    myOutput.close();
    myInput.close();
}
```

Figure 12 DBHelper code

```
public class KataKerjaUtama {
    Integer id;
    String v1;
    String v2;
    String v3;
    String ving;
    String v1s;
    String arti;

    public KataKerjaUtama() {
    }
    public KataKerjaUtama(int id, String v1, String v2, String v3, String ving, String v1s,
    String arti) {
        this.id = id;
        this.v1 = v1;
        this.v2 = v2;
        this.v3 = v3;
        this.ving = ving;
        this.v1s = v1s;
        this.arti = arti;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getV1() {
        return v1;
    }
    public void setV1(String v1) {
        this.v1 = v1;
    }
    public String getV2() {
        return v2;
    }
    public void setV2(String v2) {
        this.v2 = v2;
    }
    public String getV3() {
        return v3;
    }
    public void setV3(String v3) {
```

```

        this.v3 = v3;
    }
    public String getVing() {
        return ving;
    }
    public void setVing(String ving) {
        this.ving = ving;
    }
    public String getV1s() {
        return v1s;
    }
    public void setV1s(String v1s) {
        this.v1s = v1s;
    }
    public String getArti() {
        return arti;
    }
    public void setArti(String arti) {
        this.arti = arti;
    }
}

```

Figure 13 The script for the getter and setter for "KataKerjaUtama.java" class

```

private class KkUAdapter extends CursorAdapter {

    public KkUAdapter(Context context, Cursor c) {
        super(context, c);
    }

    @Override
    public View newView(Context cntxt, Cursor cursor, ViewGroup vg) {
        return LayoutInflater.from(cntxt).inflate(R.layout.kku, vg, false);
    }

    @Override
    public void bindView(View view, Context cntxt, Cursor cursor) {
        TextView v1 = (TextView) view.findViewById(R.id.v1);
        TextView v2 = (TextView) view.findViewById(R.id.v2);
        TextView v3 = (TextView) view.findViewById(R.id.v3);
        TextView ving = (TextView) view.findViewById(R.id.ving);
        TextView v1s = (TextView) view.findViewById(R.id.v1_s);
        TextView arti = (TextView) view.findViewById(R.id.arti);

        v1.setText(cursor.getString(cursor.getColumnIndex("v1")));
        v2.setText(cursor.getString(cursor.getColumnIndex("v2")));
        v3.setText(cursor.getString(cursor.getColumnIndex("v3")));
        ving.setText(cursor.getString(cursor.getColumnIndex("ving")));
        v1s.setText(cursor.getString(cursor.getColumnIndex("v1_s")));
        arti.setText(cursor.getString(cursor.getColumnIndex("arti")));
    }
}

```

Figure 14 The script from the class: "main activity KataKerjaUtamaActivity.java"

```

<application
    android:allowBackup="true"
    android:icon="@drawable/Logo"
    android:label="@string/app_name"
    android:theme="@style/AppTheme">
    <activity
        android:name=".SplashActivity"
        android:label="@string/app_name"
        android:screenOrientation="Landscape"
    android:theme="@android:style/Theme.Black.NoTitleBar.Fullscreen" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".MenuUtamaActivity"
        android:screenOrientation="Landscape" >
    </activity>
</application>

```

Figure 15 Script from AndroidManifest.xml.

4. TESTINGS

Table 17 shows the evaluation of the test runs of the application on the devices shown on Table 15. Seven aspects are being evaluated. “√” indicates a performance that is according to the implementation’s design, while “-“ indicates failure.

Table 17 Test Result of the Application on the Devices on Table 15

Aspects	Device				
	#1	#2	#3	#4	#5
The display’s color is as intended in the design.	√	√	√	√	√
The list item, list view, text view, edit view, and tab host buttons’s positions are as intended in the design.	√	√	√	√	√
The list item, list view, text view, edit view, and tab host buttons’s functions are as intended in the design.	√	√	√	√	√
The appearance of data on list item, list view, text view, edit view, and tab host are as intended in the design.	√	√	√	√	√
Random data is as intended in the design.	√	√	√	√	√

The blink effect for data display functions well.	√	√	√	√	√
Transition between displays work well and fast.	√	√	√	√	√

The application was tested on a sample of 30 students at UKRIM, and the result of the questionnaire as shown on Table 16 can be seen on Table 18.

Table 18
Survey Result

No.	Question	SA	A	N	D	SD
P1	The lessons explain simple statements and the fifteen kinds of tenses in the English language clearly.	17	13	0	0	0
P2	Exercises that focus on verb conjugations in simple statements as presented in the lessons help users in learning English tenses effectively and efficiently:	21	8	1	0	0
	2.a. Users are assisted in choosing the correct tenses.	19	11	0	0	0
	2.b. Users are assisted in choosing the correct verb conjugation that corresponds with a certain tense.	16	13	1	0	0
	2.c. Users are assisted to understand, that sometimes a verb consists of an auxiliary verb and a main verb.	14	15	1	0	0
	2.d. Users are assisted to understand the five widely used patterns of simple statements in the English language.	21	10	0	0	0
	2.e. Users are assisted in making a correct simple statement in English.	19	11	0	0	0
P3	The logical organization of menus and submenus makes it easy for the users to navigate through the application.	16	14	0	0	0
P4	The display is attractive.	11	17	2	0	0
P5	Access to the contents in each menu is fairly fast.	17	13	0	0	0
Total		170	125	5	0	0
Total sum		300				
Percentage		56,67%	41,67%	1,66%	0%	0%

By treating answers SA (Strongly Agree) and A (Agree) as positive answers, while N (Neither agree nor disagree), D (Disagree), and SD (Strongly Disagree) as negative answers, we get an overwhelming 98.34% of our samples responding positively to the application. We can safely deduce that we can accept our alternate hypothesis, that an Android based TiPS lessons application that focuses on verb conjugations helps Indonesian students in learning English grammar

5. CONCLUSION

An Android based application for teaching English grammar that focuses on step 3 of the TiPS method, i.e. the verb conjugations has been developed. Test runs on several smartphone devices, and testing them on a number of samples results led to the success in accepting our alternative hypothesis that an Android based TiPS lessons application that focuses on verb conjugations helps Indonesian students in learning English grammar.

We recommend a further study that can be done to better the tools in teaching English grammar to Indonesian students utilizing the concept of TiPS and using current technologies. First, this application can be made better by incorporating sound to make it more attractive as well as to assist students not only in grammar, but also in listening. Second, further study can be done to enlarge TiPS to include compound statements.

ACKNOWLEDGEMENT

The authors would like to acknowledge and thank Universitas Kristen Immanuel for the financial aid given for this study.

REFERENCES

- [1] StatCounter GlobalStats, 2015, Top 8 Mobile Operating Systems in Indonesia from Jan 2014 to June 2015, http://gs.statcounter.com/#mobile_os-ID-monthly-201401-201506 , accessed: 05-June-2015.
- [2] InfoWorld, 2015, The rise and rise of open source, <http://www.infoworld.com/article/2914643/open-source-software/rise-and-rise-of-open-source.html> , accessed: 30-May-2015.
- [3] Ardita, F. 2013, *Belajar 16 Tenses dari Nol Sampai Jago*, Buku Pintar, Yogyakarta
- [4] Maedjaja, F., 2014, *Catatan Kuliah Bahasa Inggris*, Program Studi Teknik Informatika, Universitas Kristen Immanuel, Yogyakarta
- [5] Maedjaja, F. and Zebua, A., 2015, *Web Based Application of TiPS, a Programmable Method of Teaching Simple Statement Grammar in the English Language to Indonesians*, *InFact*, Vol 01, No 02, Universitas Kristen Immanuel